Cycle Bases of Graphs and Spanning Trees with Many Leaves

Complexity Results on Planar and Regular Graphs

Von der Fakultät für Mathematik, Naturwissenschaften und Informatik
der Brandenburgischen Technischen Universität Cottbus - Senftenberg
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Dr. rer. nat.)

genehmigte Dissertation

vorgelegt von

Diplom-Mathematiker
Alexander Reich
geboren am 29.10.1982 in Lübben

Gutachter: Prof. Dr. Ekkehard G. Köhler

Gutachter: Prof. Dr. Rolf H. Möhring

Gutachter: Prof. Dr. Romeo Rizzi

Tag der mündlichen Prüfung: 15. April 2014

Abstract

A *cycle basis* of a graph is a basis of its cycle space, the vector space which is spanned by the cycles of the graph. Practical applications for cycle bases are for example the optimization of periodic timetables, electrical engineering, and chemistry. Often, cycle bases belong to the input of algorithms concerning these fields. In these cases, the running time of the algorithm can depend on the size of the given cycle basis. In this thesis, we study the complexity of finding minimum cycle bases of several types on different graph classes. As a main result, we show that the problem of minimizing strictly fundamental cycle bases on planar graphs is $\mathcal{NP}$-complete. We also give a similarly structured proof for problem of finding a maximum leaf spanning tree on the very restricted class of cubic planar graphs. Additionally, we show that this problem is $\mathcal{APX}$-complete on k-regular graphs for odd k greater than 3. Furthermore, we classify types of robust cycle bases and study their relationship to fundamental cycle bases.

Zusammenfassung

Eine *Kreisbasis* eines Graphen ist eine Basis des Zyklenraumes des Graphen, also des Vektorraumes, der von den Kreisen des Graphen aufgespannt wird. Praktische Anwendungsgebiete von Kreisbasen finden sich zum Beispiel bei der Optimierung von Taktfahrplänen, in der Elektrotechnik und der Chemie. Oft gehören Kreisbasen zur Eingabe von Algorithmen aus diesen Gebieten. In diesen Fällen kann die Laufzeit der Algorithmen von der Größe der eingegebenen Kreisbasis abhängen. In dieser Arbeit untersuchen wir die Komplexität, minimale Kreisbasen unterschiedlicher Typen auf verschiedenen Graphenklassen zu finden. Als ein Hauptergebnis zeigen wir, dass das Problem minimale streng fundamentale Kreisbasen von planaren Graphen zu finden $\mathcal{NP}$-vollständig ist. Wir geben einen ähnlich strukturierten Beweis für das Problem, auf der sehr eingeschränkten Graphenklasse der kubischen planaren Graphen einen spannenden Baum mit der maximalen Anzahl an Blättern zu finden. Zusätzlich zeigen wir, dass dieses Problem auf k-regulären Graphen für ungerade k größer als 3 $\mathcal{APX}$-vollständig ist. Ein weiteres Thema der Arbeit ist die Klassifizierung von Typen robuster Kreisbasen und ihre Beziehung zu fundamentalen Kreisbasen.

Acknowledgements

Since my first contact with cycle bases of graphs as a student in 2006 I met a number of people who raised my interest in this topic further and further. At this place, I want to take this opportunity of thanking them for all of their valuable advices and inspirations in a chronological order. First of all, I wish to thank Ekkehard Köhler for organizing my internship at the Technical University of Berlin. The same thanks must also go to Christian Liebchen and Gregor Wünsch for their intensive supervision during that time. Also my diploma thesis and my first publication arose from this internship.

It had been a pleasure for me to participate in the *Workshop on Cycle and Cut Bases* which took place at the Eberhard Karls University, Tübingen in May 2008. In these days, I became truly inspired in many problems concerning cycle bases which had been previously unknown to me. Thank you to all organizers of this workshop, especially to Katharina Zweig, and again to Christian Liebchen, who had drawn my attention to this workshop.

Thank you also to the organizers of the 9th *SEG Workshop über Kombinatorik, Graphentheorie und Algorithmen* which was held at the Chemnitz University of Technology in June 2011. In August 2012, FRICO took place at the Zuse Institute Berlin. At this workshop, I could present some results of this thesis, especially on spanning trees with many leaves. Thanks to the organizers of FRICO 2012.

Now, as the thesis is accomplished, I am especially grateful to my supervisor Ekkehard Köhler. Thank you for your support, encouragement, motivation, and all the helpful suggestions which I received from you. I also want to thank my assessors Rolf H. Möhring and Romeo Rizzi.

I do not want to forget to thank all members of staff on the third floor in the Main Building of BTU Cottbus for the supply of an efficient, motivating, and simply optimum working environment. In particular, I want to thank Martin Strehler and my former roommate Harry Schülzke. Furthermore, I want to thank all proofreaders of the manuscript, in particular Martin and Randolf, as well as Uwe for his technical support. A really special thank you is also due to Diana Hübner.

Finally, I want to express my gratitude to my family, my friends in Sellendorf, and to my girlfriend Jana Paulick.

Cottbus, October 2013 Alexander Reich

Contents

Introduction

Graph drawing. Chemistry. Electrical engineering. Category theory. Periodic timetabling. What do these areas have in common? How does category theory matches up with this list? Beyond others, these fields are applications for different classes of cycle bases, the main topic of this thesis.

From an etymological point of view, the word *cycle* is derived from the Late Latin *cyclus* respectively from the Greek word κύκλος ([96]) and means "any complete round or series of occurrences that repeats or is repeated" ([35]). Besides cycles, there are three further terms used in this thesis to describe mathematical objects which are round in a sense. In our meanings, *circulation* is essentially the same as a cycle, while a *circuit* comes along with a simpler structure. Finally, by a *circle* we mean what each elementary-school pupil ought to know, namely the set of all points in a plane that have the same distance to a given point.

Anyway, circles will not play a major role. Rather, we will deal with circuits and cycles in graphs. The set of all cycles in a graph forms a vector space. And this vector space, which is called the cycle space in this case, has a basis—the cycle basis.

The investigation of cycle bases has addressed many researchers, especially in the last 15 years. This is also reflected by numerous publications on cycle bases of graphs. The authors of these papers considered cycle bases not only from a practical point. Additionally, they investigated structural properties of different classes of cycle bases. Further lines of research reached from exact and approximation algorithms for the computation of minimum cycle bases restricted to specified graph classes, over the classification of different types of cycle bases, to statements on the complexity of computing a minimum cycle basis of a special class, just to name a few. Intelligibly, this led to many continuing questions and further open problems. One of the main pretensions of this thesis is to localize some of these unresolved problems and to solve them admittedly in parts.

During the research on cycle bases, we had been confronted with another area in addition to cycle bases. Since it deals with trees, which do not contain cycles, it seems to have not much in common with cycle bases, at a first view. However, this view changes when we are concerned with strictly fundamental cycle bases. But what we are actually talking about is the problem of finding a spanning tree of a graph that maximizes the number of leaves. Also this problem attracted the attention of many researchers, who specialized for instance in approximation algorithms for general and for cubic graphs, in particular. We came across this problem due to the similarity between the completeness proofs for this problem restricted to planar and cubic graphs on one hand, and for the problem of finding a minimum strictly fundamental cycle basis of a planar graph on the other hand.

In the next paragraphs, we outline the thesis and motivate the sense of reading the thesis from cover to cover. Moreover, one of the references, a survey on cycle bases, stands out from the others, thus, it is mentioned in an extra paragraph.

Outline of the Thesis. Chapter 1 is used to summarize basic definitions and to fix our notation which is used in this thesis. The chapter is divided into three sections, where the first one deals with elementary graph theoretical concepts. The second section contains the definitions on cycles and cycle bases for directed and undirected graphs. Additionally, we discuss advantages and disadvantages of different viewpoints of cycles. In the third section, notions on complexity and approximation are given. In particular, we specify the terms decision and optimization problem, the Landau notation, computational complexity, approximation algorithms and schemes, complexity classes, as well as reductions and completeness.

Chapter 2 treats the Maximum Leaf Spanning Tree Problem. One main result in this chapter is the proof of $\mathcal{NP}$-completeness of this problem when we restrict to planar and cubic graphs. The proof is done via a reduction from the Exact Cover by 3-Sets Problem for planar graphs. After some notes on the embedding of instances of this problem which is essential for the proof, we proceed with an exhaustive description of the reduction. After this, the proof is further adapted to graphs which are additionally biconnected. The second main result is the $\mathcal{APX}$-completeness of the problem for 5-regular graphs. This proof is extended to graphs with an arbitrary odd regularity greater than 5. As a positive result, we deduce some exact and one approximation algorithm for selected graph classes.

Strictly fundamental cycle bases are the topic of Chapter 3. It begins with stringent definitions and some technical notes, after what follows a detailed view of strictly fundamental cycle bases on planar graphs. We carry on with the Minimum Strictly Fundamental Cycle Basis Problem and illustrate its relationship to the Optimum Communication Spanning Tree Problem. Thereafter, strictly fundamental cycle bases on planar graphs with non-metric weighting functions are considered. After this, we proceed with the main result of this chapter, namely the $\mathcal{NP}$-completeness of the Strictly Fundamental Cycle Basis Problem for planar graphs. Further considerations of this chapter are strictly fundamental cycle bases on weighted outerplanar graphs and on weighted cycle root graphs, i.e. graphs closely related to the duals of outerplanar graphs.

The research on robust cycle bases is carried on in Chapter 4. We figure out the four established types of robust cycle bases, i.e. cyclically robust, strictly robust, quasi-robust, and strictly quasi-robust cycle bases. We draw a map which respects trivially outcoming inclusions. It is shown with suitable examples that no two of these classes coincide on general graphs. Each of these examples provides a cycle basis which is minimum on its graph. A second issue of this chapter is the comparison of robust cycle bases and fundamental cycle bases, what had been initiated in [67]. Except one, also these examples constitute minimum cycle bases on their graphs.

Chapter 5 collects results on further classes of cycle bases. As a new class, we introduce p-bases as a generalization of 2-bases. We investigate for which p such bases exist or not. In the section about totally unimodular cycle bases different definitions are related to each other. Moreover, totally unimodular cycle bases are compared with weakly fundamental cycle bases, where we discover a weak kind of a hierarchical structure. The last section is about integral cycle bases, a class of practical interest. After a description of the PESP for cyclic timetabling, we turn our focus on integral cycle bases without simple cycles and on the coefficients of linear combinations for circuits.

How to Read this Thesis? We suggest to read the chapter in the order of their appearance in this thesis. Clearly, Chapter 1 on the preliminaries should be read at first. Chapter 5 summarizes several results on further classes of cycle bases. It uses some issues of Chapter 3 on strictly fundamental cycle bases and thus ought to be read after it. Moreover, Chapter 2 and Chapter 3 contain two similarly structured proofs. Since the former proof is less challenging it should be read prior. This is also the chronological order of the development of the proofs. Finally, Chapter 4 on robust cycle bases can be read independently of the other ones.

From Chapter 2 on, each chapter starts with an outline and a summary of our contribution. All of these chapters end with some conclusions and quotations of questions which remained open. Chapter 2 to 4 contain introductions with some historical notes and descriptions of practical applications. In the last chapter, these parts are partially shifted into the sections.

As usually, the end of a proof is marked by the symbol $\square$. In addition, the thesis is loaded with examples, especially in Chapter 4 on robust cycle bases. To mark the end of an example, we use the symbol $\lozenge$.

The Survey. A great help and inspiration for the preparation of this thesis had been the survey [64], written by Kavitha, Liebchen, Mehlhorn, Michail, Rizzi, Ueckerdt, and Zweig. Therein, the authors "surveyed structural, algorithmic, and complexity-theoretical results and compiled a list of open problems.". From this list of 15 open problems, we were able to solve three of them partially (Open Problems 3, 4 and 6) and one completely (Open Problem 14). Since this survey is mentioned very frequently, we sometimes pass a correct citing up and simply speak about *the Survey*.

Chapter 1

Preliminaries

This first chapter is dedicated to prepare the elementary concepts which are used in this thesis. The basic graph theoretic definitions are collected in Section 1.1. If any further graph theoretical terms are used in this thesis but not declared in this section, we refer to standard textbooks such as [17,18,36,123]. On the other hand, terms which are defined in this section but do not appear anywhere else, can be ignored. Clearly, a cycle *is* a basic graph theoretical concept. In spite of this, the stringent definition of a cycle is moved to Section 1.2. In this section, we also discuss different viewpoints of cycles and give precise definitions of the cycle spaces for undirected and directed graphs. The section is closed with a very short illustration of the cycle matrix and its determinant. Section 1.3 provides a brief summary of the used concepts associated with complexity and approximability. It contains some standard material from the field of theoretical computer science, collected from several textbooks—for concrete references look on site. Beyond the overview of problem types, the Landau notation, some words to computational complexity, and approximation algorithms, we emphasized the types of reductions and completeness which occur in this thesis.

1.1 Graph Theory

An *undirected graph* $G = (V, E)$ is a pair consisting of a finite set V of *vertices* or *nodes* and a set of *edges* $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$. Throughout this thesis we fix $n = |V|$ and $m = |E|$. When we keep busy with more than one graph, we may write $V(G)$ and $E(G)$ to denote the vertex and edge set of the graph G. Per definition, an edge is a 2-element subset of V. Nevertheless, we usually write $e = uv$ instead of $e = \{u, v\}$ to denote an edge. Two nodes u and v are *adjacent* if there is an edge $e = uv \in E$. The vertices u and v are called the *end nodes* of e in this case. If all vertices are pairwise adjacent, the graph is called *complete*. The *neighborhood* $N(v)$ is the set of all vertices adjacent to v, the *closed neighborhood* is denoted by $N[v] := N(v) \cup \{v\}$. A vertex v and an edge e are *incident* if $v \in e$.

In a *multigraph*, the edge set $E \subseteq \{\{u,v\} \mid u,v \in V\}$ is allowed to be a multiset. Additionally, also the edges themselves can be multisets, i.e. edges $\{u,u\}$ are possible. They are called *loops*. Two or more edges are termed *parallel* if they are incident to the same (multi-)set of nodes. The term *simple graph* is used when the graphs defined in the first paragraph shall be set apart from multigraphs.

The *degree* $\deg(v)$ of a node v is the number of its incident edges. A graph is *k-regular* if $\deg(v) = k$ for all $v \in V$; a graph is *cubic* if it is 3-regular. For a graph $G = (V,E)$ and vertices $u,v \in V$, define $G + uv := (V, E \cup \{uv\})$, $G \setminus uv := (V, E \setminus \{uv\})$, and $G \setminus v := (V \setminus \{v\}, E \setminus \{e \in E \mid v \in e\})$. If $e = uv \in E$, then G/e is the graph which arises by *contracting* e to a new vertex v_e. More precisely, $G/e := (V \setminus \{u,v\} \cup \{v_e\}, \{xy \in E \mid \{x,y\} \cap \{u,v\} = \emptyset\} \cup \{v_e x \mid x \in N(u) \cup N(v) \setminus \{u,v\}\})$.

A *weighted graph* $G = (V,E)$ additionally provides a weighting function $w : E \to \mathbb{K}$, where $\mathbb{K}$ is mostly $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, or $\mathbb{N} \cup 0$. Two graphs $G = (V,E)$ and $G' = (V',E')$ with weighting functions w and w' are *isomorphic* if there exists a bijection $\varphi : V \to V'$ with $uv \in E \Leftrightarrow \varphi(u)\varphi(v) \in E'$ and with $w(uv) = w'(\varphi(u)\varphi(v))$. Sometimes, the weighting function is included into the graph itself. A weighted graph is then denoted by $G = (V,E,w)$.

A *walk* with length ℓ is a sequence $(v_0, e_1, v_1, \ldots, e_\ell, v_\ell)$ of vertices and edges with e_i being incident with v_{i-1} and v_i for $1 \leq i \leq \ell$. If all vertices of a walk are pairwise disjoint, this walk is referred to as *v_0-v_ℓ-path* or simply as *path*. The vertices v_0 and v_ℓ are called the *end nodes of the path*. A path can also be regarded as a sequence of vertices or as a sequence of edges. Since each vertex and each edge appears only once, it can also be seen as a set of edges instead of a sequence. For $u,v \in V$ denote $\mathcal{P}_{u,v}$ the set of all u-v-paths. With P_m we denote the path with m edges. The *length of a path P* is defined as $\mathrm{length}(P) := \sum_{e \in P} w(e)$. The *distance* $\mathrm{dist}_G(u,v)$ is the minimum length of a path from u to v in G. The *eccentricity* of a node v is the maximum distance of another node to v, formally $\mathrm{ex}(v) := \max_{u \in V} \mathrm{dist}(u,v)$. Now, the *diameter* of a graph $G = (V,E)$ can be defined as the maximum eccentricity over all nodes, hence as $\mathrm{diam}(G) := \max_{v \in V} \mathrm{ex}(v)$. The *distance* of a node v to a vertex set $U \subseteq V$ is defined by $\mathrm{dist}_G(v,U) := \min_{u \in U} \mathrm{dist}_G(u,v)$. An edge $e = uv$ of a weighted graph is *metric* if $w(e) = \mathrm{dist}_G(u,v)$; a graph itself is called *metric* if all its edges are metric. Often, we also skip the index G.

Let $V' \subseteq V$ and $E' \subseteq E$ for a graph $G = (V,E)$, then $G' = (V',E')$ is called a *subgraph* of G if $E' \subseteq \{uv \in E \mid u,v \in V'\}$. If $E' = \{uv \in E \mid u,v \in V'\}$, then the subgraph is *induced by V'*. Denote $G[V']$ the subgraph itself. Similarly, if $V' = \bigcup E'$, then $G' = G[E']$ is *induced by E'*. For a subset $E' \subseteq E$ we write $G \setminus E' := (V, E \setminus E')$. A *spanning subgraph* is a subgraph with $V' = V$. An induced subgraph is a *clique* if it is complete. The size of a largest clique in a graph G is the *clique number* $\omega(G)$. If G' is a subgraph of G, then G is a *supergraph* of G'. For a graph $G = (V,E)$, its *complement graph* is defined as $\overline{G} = (V, \overline{E})$, where $\overline{E} = V^2 \setminus E$.

A *directed graph* or *digraph* $D = (V, A)$ consists of a finite node set V and the set of *arcs* $A \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$. Again, we often denote $a = uv$ instead of $a = (u, v)$. It is reasonable to customize the notion of neighborhood to directed graphs, thus, $N^-(v) := \{u \mid uv \in A\}$ and $N^+(v) := \{w \mid vw \in A\}$. Further, the *in-* and the *outdegree* of a vertex v are defined as $\deg^-(v) := |N^-(v)|$ and $\deg^+(v) := |N^+(v)|$, respectively. Additionally, we set $\alpha(a) = u$ as the *tail* and $\omega(a) = v$ as the *head* of an arc $a = uv$. Although many other definitions usually also differ slightly for the directed case, this distinction is not necessary in this thesis. Thus, all other definitions above apply directly to digraphs. For example, an arc in a path of a directed graph can also be passed in opposite direction, i.e. from head to tail. The reason for this deviation is that the same shall hold for cycles in the cycle space. For a digraph $D = (V, A)$ define the *underlying graph* $G(D) := (V, \{\{u, v\} \mid (u, v) \in A\})$. On the other hand, each undirected graph G can be considered as a digraph D by orienting the edges arbitrarily. D is then called an *orientation* of G.

A graph is *connected* if there is a u-v-path for each pair $u, v \in V$, and *disconnected* otherwise. An induced subgraph which is connected and which has a maximal number of nodes is called a *connected component* or just *component*. Denote $c(G)$ the number of connected components of G. More generally, a graph $G = (V, E)$ is called k-*connected* for a $k \in \mathbb{N}$ if $k < |V|$ and if for each subset $V' \subset V$ with $|V'| < k$ the graph $G[V \setminus V']$ is connected. For $k = 2$ and $k = 3$ the terms *bi-* and *triconnected* are also in use. For a real number t, a graph G is t-*tough* if for each $k > 1$, the graph G cannot be decomposed into k components by removing less than tk vertices, i.e. $t \cdot c(G[V \setminus S]) \leq |S|$ does hold for each subset $S \subseteq V$. We call a digraph D *connected* if the underlying graph $G(D)$ is connected. All other definitions in this paragraph apply to digraphs if they do for their underlying graphs, as well.

A *tree* is a connected graph with $m = n - 1$ arcs respectively edges. Note that this definition differs from the usual one in most textbooks about graph theory, where trees are defined as connected graphs without cycles. The reason for our different definition is that we decided to define cycles later. Moreover, the orientation of the arcs of a tree does not play a role in our context. Vertices v of a tree T with $\deg_T(v) = 1$ are called *leaves*. For a graph $G = (V, E)$, a spanning subgraph $G' = (V, E')$ which is a tree is called a *spanning tree* of G. With $\ell(G, T)$ we denote the number of leaves of a spanning tree T of a graph G. The edges in $E \setminus E(T)$ are referred to as *chords of the spanning tree*.

The *incidence matrix* $\mathcal{I}(D) = (i_{jk})_{n \times m}$ of a directed graph D resp. $\mathcal{I}(G) = (i_{jk})_{n \times m}$ of an undirected graph G is defined as

$$i_{jk} = \begin{cases} 1, & \text{if } \alpha(a_k) = v_j \\ -1, & \text{if } \omega(a_k) = v_j \\ 0, & \text{otherwise,} \end{cases} \quad \text{resp.} \quad i_{jk} = \begin{cases} 1, & \text{if } v_j \in e_k \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

For the sake of completeness we also define the *adjacency matrix* $\mathcal{A}(D) = (a_{jk})_{n\times n}$ ($\mathcal{A}(G) = (a_{jk})_{n\times n}$) of a directed graph $D = (V, A)$ or an undirected graph $G = (V, E)$ as

$$a_{jk} = \begin{cases} 1, & \text{if } (v_j, v_k) \in A \text{ resp. } \{v_j, v_k\} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{1.2}$$

1.2 Cycles and Cycle Spaces

The main subject of this thesis is the concept of *cycle bases*. Thus, when investigating this topic, one should exactly know what a cycle is. During the history of cycle bases—or more generally, of graph theory—different perceptions have been developed of what a *cycle in a graph* ought to be. It turned out that the usage of different notions of cycle bases and hence of cycles in undirected and directed graphs is advantageous. One reason for this is the possibility of backward arcs in a cycle of a digraph. Since we will also use several different notations for cycles, we illustrate them below. Additionally, we discuss their advantages and disadvantages. Nevertheless, we decided to omit a clear definition of cycles at this place and catch this up below. We do this in favor to look at cycles and other substructures from an abstract and set-theoretic point of view.

Cycles as Subgraphs. Since a cycle can be viewed as a graph on its own, this notion is maybe the most intuitive one. Given a graph $G = (V, E)$, a cycle in G is just a subgraph $C = (V(C), E(C))$ with $V(C) \subseteq V$, $E(C) \subseteq E$ and a constraint for being a cycle such as $\deg(v) \equiv 0 \mod 2$ for all $v \in V$. The advantage of this model is its simplicity. However, this cannot reflect all the features of cycle bases, especially in the directed case.

Cycles as Sets of Edges. Here, a cycle is simply a set $C = \{e_1, e_2, \ldots, e_\ell\}$ with $\ell \leq m$. The cyclic structure of this set of edges in the graph $G = (V, E)$ is given if $|\{e \in C \mid v \in e\}| \equiv 0 \mod 2$ holds for all $v \in V$. With this model, it is easier to denote a cut or the sum of cycles. When dealing with spanning trees, it is usually more convenient to interpret them as sets of edges. Concerning cycles, this model has similar problems as the first one.

Cycles as Sequences of Edges or Vertices. From this perspective, a cycle is similarly defined as a walk in Section 1.1, i.e. $C = (v_0, e_1, \ldots, e_\ell, v_\ell = v_0)$. Here, the cyclic structure can be expressed in the same way as for the perception of cycles as edge sets. It is also possible to use sequences consisting exclusively of vertices or edges, respectively. This model of cycles as sequences has mainly two advantages. On one hand, if the considered graph is directed, every *integral cycle*[1] is representable. And on the other hand, the ordering of the edges and vertices is explicitly given. Problems occur when we try to operate with cycles in this representation, e.g. to add them.

[1] See Definition 1.2.

Cycles as Vectors. Since the cycle space is a vector space, this should be our preferred model for cycles. For a fixed order of the arcs, indicated by their indices, a cycle is a column vector $C = (C_{a_1}, C_{a_2}, \ldots, C_{a_m})^\mathsf{T}$. For the constraints of being a cycle see Definition 1.1. A significant advantage over the other models is the possibility that C_{a_i} can take different values. Moreover, this is the only model where actually every element of the cycle space of a directed graph is representable. Sometimes, we also write $C(a_i)$ instead of C_{a_i}. When regarding cycles as vectors, the *inner product* of two cycles C and C' is defined as usual, i.e.

$$\langle C, C' \rangle := \sum_{i=1}^{m} C_{a_i} C'_{a_i}. \tag{1.3}$$

Clearly, all four perspectives above can directly be carried over to arbitrary subgraphs such as trees, cuts, or paths. Furthermore, it is straightforward to switch between the first two notions. To see this, let C_G be a cycle given as a subgraph of a graph G and let C_S be a cycle represented as a set of edges. Then we can simply set $C_S = E(C_G)$ and $C_G = G[C_S]$. Moreover, assume that C_F is a cycle given as a sequence of edges. The access to elements of this sequence is accomplished in regarding the sequence as a discrete function $C_F : \{1, \ldots, \ell\} \to E$. Now, the cycle as set of edges can be expressed as $C_S = \{e \in E \mid \exists i \in \{1, \ldots, \ell\} : C_F(i) = e\}$. However, the order of the edges is lost in this case. In addition, if C_F is a cycle in a directed graph and several edges appear more than once in C_F then C_S as defined above does not constitute a cycle anymore. The relationship to cycles as vectors is accomplished by the *support*, which is defined below.

Which of the perspectives is used, depends on the context. Note that we will not explicitly state each time which characteristic is currently in use. We proceed with formal definitions of cycles and cycle spaces, where the same names are used, although there are differences between the directed and the undirected case.

Definition 1.1 (cycle, cycle space, edge space, support). *A* cycle *in an undirected graph $G = (V, E)$ is a vector $C \in \mathrm{GF}(2)^{|E|}$ with*

$$\sum_{u \in N(v)} C_{uv} \;=\; 0 \quad \textit{for all } v \in V. \tag{1.4}$$

In a digraph $D = (V, A)$, a cycle *is a vector $C \in \mathbb{K}^{|A|}$ for which*

$$\sum_{u \in N^-(v)} C_{uv} \;=\; \sum_{w \in N^+(v)} C_{vw} \tag{1.5}$$

holds for all $v \in V$ and $\mathbb{K}$ is a field, usually $\mathbb{Q}$ or $\mathbb{R}$.

The sets $\mathcal{C}(G) = \{C \mid C \text{ is cycle in } G\}$ and $\mathcal{C}(D) = \{C \mid C \text{ is cycle in } D\}$ form vector subspaces of $\mathrm{GF}(2)^{|E|}$ resp. of $\mathbb{K}^{|A|}$, the cycle spaces *of G and D. The spaces $\mathrm{GF}(2)^{|E|}$ and $\mathbb{K}^{|A|}$ themselves are referred to as the* edge space *and the* arc space*. The* support $\mathrm{supp}(C)$ *of a cycle or any other vector C is the set of edges e with $C_e \neq 0$, respectively the set of arcs a with $C_a \neq 0$.*

We should distinguish between several special cycle types in a digraph.

Definition 1.2 (integral cycle, simple cycle, circuit, projection). *Let C be a cycle in a directed graph $D = (V, A)$. Then C is called*

- integral cycle*, if $C_a \in \mathbb{N}$ holds for all $a \in A$,*
- simple cycle*, if $C_a \in \{-1, 0, +1\}$ holds for all $a \in A$,*
- circuit*, if $D[\operatorname{supp}(C)] \setminus (A \setminus \operatorname{supp}(C))$ is connected and 2-regular.*

We usually denote an integral cycle by I, a simple cycle by S, and a circuit by C. A cycle in an undirected graph is a circuit *if $G[\operatorname{supp}(C)] \setminus (E \setminus \operatorname{supp}(C))$ is connected and 2-regular. For a simple cycle $S = (S_{a_1}, \dots, S_{a_m})^\mathsf{T}$ in a digraph D define the* projection *$\pi(S) := (|S_{a_1}|, \dots, |S_{a_m}|)^\mathsf{T}$ of S onto the underlying undirected graph $G(D)$.*

Definition 1.3 (Hamiltonian circuit). *A circuit C is a* Hamiltonian circuit *in a graph G if $G[\operatorname{supp}(C)] = G$, i.e. if C contains all vertices of G.*

Definition 1.4 (cycle basis, cyclomatic number, basic circuit/cycle). *A basis of the cycle space is a* cycle basis*. An element of the cycle space is called* basic circuit *or* basic cycle*. The* cyclomatic number *or the* nullity *is the dimension of the cycle space, its value is $\nu(G) = m - n + c(G)$. If there are no confusions concerning the considered graph, we often simply denote the cyclomatic number by ν.*

A cycle basis $B = \{C_1, \dots, C_\nu\}$ of the cycle space $\mathcal{C}(D)$ of a directed graph D is called *directed cycle basis* if there are no further restrictions on B. If also the projection $\pi(B) := \{\pi(C_1), \dots, \pi(C_\nu)\}$ is a cycle basis of the cycle space of the underlying graph $G(D)$, the basis B is referred to as *undirected cycle basis*. For a separating example which show that both notions are different, see Figure 1.1 which is the same as Figure 5 in [64]. More precisely, it shows a graph and a directed cycle basis where each arc is contained in exactly two basic circuits. Clearly, their projections are linearly dependent.

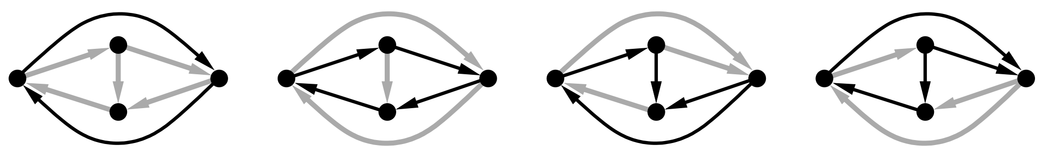


Figure 1.1: A graph and a directed cycle basis (black arcs) which is not undirected.

Sometimes, we also speak about the *cycle basis of a graph* for short. For the sake of completeness, for a directed graph D the *co-cycle space* or the *cut space* is the orthogonal vector subspace of the cycle space with respect to the inner product. Its dimension or *rank* is $\rho = \rho(D) = n - 1 - c(D)$. The elements of the cut space are the *cuts* of D. More descriptively expressed, a cut in a digraph can be seen as a (electric) tension, i.e. as potential difference of some potential function on the vertex set. Formally speaking, a vector S is a directed cut, if there is a function $\pi : V \to \mathbb{R}$ such that $S_{uv} = \pi(v) - \pi(u)$. The support of a cut is a set of arcs whose removal disconnects the graph.

From an algebraic point of view, the cycle and the edge space of a graph D can easily be described in terms of the incidence matrix $\mathcal{I}(D)$. More precisely, the cycle space is the kernel of $\mathcal{I}(D)$, while the cut space is the image of $\mathcal{I}(D)^\mathsf{T}$. This also motivates the terms *nullity* and *rank* for their dimensions.

Note that here cycles are defined very generally. For a cycle C in an undirected graph G, the graph $G[\mathrm{supp}(C)]$ may be disconnected. It is only required that each of its vertices has an even degree. In particular, degrees of four or more are possible. In the directed case, a cycle can be viewed as a circulation. That's why $\mathcal{C}(D)$ sometimes is called *circulation space* ([18]) or *flow space* ([55]).

Definition 1.5 (weight, size, and length of a cycle (basis), girth). *The* weight *of a cycle C in a weighted digraph $D = (V, A)$ is defined as*

$$w(C) = \sum_{a \in A} |C_a| w(a). \tag{1.6}$$

If the graph is unweighted, $w(C)$ is also called the size *or the* length *of C and denoted by $|C|$. The minimum size of a circuit in an unweighted (di-)graph G is called the* girth *of G. The* weight *$\Phi(B)$ of a cycle basis B is defined as*

$$\Phi(B) := \sum_{C \in B} w(C). \tag{1.7}$$

Remember that a cycle C is a vector in $\mathrm{GF}(2)^{|E|}$ for the case of an undirected graph. The definitions above apply to undirected graphs, when the entries C_e of a cycle C and $\sum$ are regarded as elements and as the sum in $\mathbb{Q}$ or $\mathbb{R}$ instead of GF(2).

Minimization of the size respectively the weight $\Phi(B)$ of a cycle basis B is a main topic in this thesis. Thus, for a graph G denote by $\Phi(G)$ the minimum value of $\Phi(B)$ over all cycle bases B of $\mathcal{C}(G)$. For a digraph D, the invariant $\Phi(D)$ is defined as the minimum value over all cycle bases which contain only integral cycles. Sometimes, when we do not want to specify a particular graph, we also write simply Φ.

For a given cycle basis $B = \{C_1, \ldots, C_\nu\}$, the corresponding *cycle matrix* $\Gamma = (\gamma_{jk})_{m \times \nu}$ is defined by setting $\gamma_{jk} = C_k(a_j)$. The *determinant of a cycle basis B* can be defined as $\det(B) = |\det(\Gamma')|$, where the *reduced cycle matrix* Γ' arises from Γ by deleting the rows that correspond to the arcs of a spanning tree. Clearly, the reduced cycle matrix is not unique. In contrast, the determinant of a cycle basis is always uniquely determined and does not depend on the choice of the spanning tree. Several classes of cycle bases can be characterized by looking at their determinants, see e.g. the Survey. Note that although the cycle matrix suggests an ordering of the cycles in a basis, we prefer to denote a cycle basis as a set.

1.3 Complexity and Approximation

The task of this section is to give an introduction to the fields of complexity and approximation. Since these are very broad areas, we can only give an overview. Actually, the function of this section is to fix our notations. The description of the concepts presented here are borrowed from different theoretical computer science textbooks, accomplished to a unique notation. If not explicitly stated differently, we refer for instance to [53, 59, 71].

Decision and Optimization Problems. In this thesis we consider mainly two types of problems. In a *decision problem* we are given a problem instance as input and a decision that corresponds to this input. The answer may either be "YES" or "NO". An *optimization problem* asks for a feasible solution with an optimal value. Below, both notions are defined in a more formal way.

Definition 1.6 (decision problem). *A* decision problem *P is a pair $(\mathcal{I}, S)$, where $\mathcal{I}$ is the set of* instances *of P and $S : \mathcal{I} \to \{$"YES", "NO"$\}$ is a function. Solving P on an instance $I \in \mathcal{I}$ means to decide whether $S(I) =$ "YES" or $S(I) =$ "NO" holds.*

Definition 1.7 (optimization problem, objective function). *An* optimization problem *P is a 4-tuple $(\mathcal{I}, S, c, \mathrm{opt})$, where $\mathcal{I}$ is the set of* instances *of P and $S(I)$ is the set of* feasible solutions *of an instance $I \in \mathcal{I}$. The function $c : \mathcal{I} \times S \to \mathbb{N}$ is called the* objective function*, and $\mathrm{opt} \in \{\max, \min\}$. Solving P on an instance $I \in \mathcal{I}$ means to find a solution $s \in S(I)$ which maximizes respectively minimizes $c(I, s)$. We may abbreviate $\mathrm{opt}(I) = \mathrm{opt}\{c(I, s) \mid s \in S(I)\}$.*

Remark 1.8. *It is easy to transform an optimization problem into a decision problem. For that, an additional parameter k is appended to the instance $I \in \mathcal{I}$ of the optimization problem P. The question to the corresponding decision problem is whether there is a solution $s \in S(I)$ with $c(I, s) \geq k$ $(\leq k)$ if P is a maximization (minimization) problem.*

The Landau Notation. When analyzing algorithms, it is rather hard to imagine to do this without the Landau notation, which is used to describe the asymptotic behaviour of functions and classes of functions. Actually, it is a family of five symbols. They are mentioned at this place, because the definitions differ from each other in the references. Moreover, we want to give some statements about confusing habits when using this notation.

Definition 1.9 (Landau notation). *Let $g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be a function. Then* Landau's symbols *are defined as*

- $\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \exists c \in \mathbb{R}_{>0}, n_0 \in \mathbb{N} \;\; \forall n \geq n_0 : f(n) \leq c \cdot g(n)\}$,
- $\Omega(g) = \{f : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \exists c \in \mathbb{R}_{>0}, n_0 \in \mathbb{N} \;\; \forall n \geq n_0 : f(n) \geq c \cdot g(n)\}$,

- $\Theta(g) = \mathcal{O}(g) \cap \Omega(g)$,
- $o(g) = \{f : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \forall c > 0 \;\; \exists n_0 \in \mathbb{N} \;\; \forall n \geq n_0 : f(n) < c \cdot g(n)\}$,
- $\omega(g) = \{f : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \forall c > 0 \;\; \exists n_0 \in \mathbb{N} \;\; \forall n \geq n_0 : f(n) > c \cdot g(n)\}$.

A function g is called *superpolynomial* if $g \in \omega(f)$ for all polynomials f. If g is superpolynomial then the notation $\mathcal{O}^*(g) = \mathcal{O}(pg)$ for some polynomial p is also in use; furthermore, it is $\tilde{\mathcal{O}}(g) = \mathcal{O}(gh)$ with $h(n) \in \mathcal{O}(\log^k(n))$ for some $k \geq 1$. Although classes respectively sets of functions are defined, often one sees confusing notations like "$f = \mathcal{O}(g)$". Anyway, we abide by the more convenient notation "$f \in \mathcal{O}(g)$". Another notational quality of Landau's symbols is to use the mapping rule instead of the function's name. Exemplary, we write $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(f)$ for the function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ with $n \mapsto n \log n$. Since we deal with graphs which have n vertices *and* m edges, we usually extend the definitions of Landau's symbols for two variables by replacing the domains of the functions f and g with $\mathbb{N} \times \mathbb{N}$. In addition, the term "$\forall n \geq n_0$" is substituted by "$\forall n, m \geq n_0$" in each case.

Beside the indication of running times of algorithms—see the next paragraph—Landau's symbols are also used to state a priori sizes of solutions, e.g. of minimum cycle bases. For instance, if $\Phi(B) \leq f(n, m)$ holds for each minimum cycle bases B (of a specified type) on all graphs (in a specified class) with n vertices and m edges for a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$, we denote $\Phi(B) \in \mathcal{O}(f)$ or write that "the size of a minimum basis is in $\mathcal{O}(f)$". Moreover, Landau's symbols may also be a part of a term. For instance, if $f(n) = g(n) + h(n)$ where $h \in o(n)$ is a function which is not specified in more detail, we also write $f(n) = g(n) + o(n)$.

Computational Complexity. Considering the complexity of an algorithm, the usual measurements are *time* and *space*. However, we are mainly concerned with time complexity.

To determine the *running time of an algorithm*, all elementary operations such as additions, comparisons, or assignments, are counted. Clearly, this running time of an algorithm depends on the input respectively the instance. Thus, the running time can be regarded as a function of the input. The instance is given to the algorithm in a binary encoding. The *size* $|I|$ of an instance I is the length of this encoding. Now we are able to define the *time complexity function* of an algorithm as the function which maps an input size to the maximum running time of this algorithm on an input of this size. A *polynomial time algorithm* is an algorithm whose time complexity function is in $\mathcal{O}(p)$ for some polynomial p.

Approximation Algorithms and Schemes. In this paragraph we sketch the definition of approximation algorithms. Furthermore, we define polynomial time approximation schemes, although they appear rather rarely in this thesis. Since we only deal with relative approximation we skip the declarations of algorithms with an absolute performance guarantee, as well as of fully polynomial-time approximation schemes. For the correlation of algorithms with relative performance guarantee with these notions see e.g. [7].

Definition 1.10 (approximation algorithm, performance guarantee/ratio). *Let $P = (\mathcal{I}, S, c, \mathrm{opt})$ be an optimization problem as in Definition 1.7 and $k \in (1, \infty)$. An* approximation algorithm *with* performance guarantee k *is a polynomial time algorithm that computes for each instance $I \in \mathcal{I}$ a solution $s \in S(I)$ with*

$$R_P(I, s) := \max\left\{\frac{c(I,s)}{\mathrm{opt}(I)}, \frac{\mathrm{opt}(I)}{c(I,s)}\right\} \leq k. \tag{1.8}$$

The value $R_P(I, s)$ is also called the performance ratio *of the solution s with respect to a given instance I of a problem P.*

Note that there are also algorithms with a non-constant performance guarantee. If we do not explicitly declare something else, we refer to an algorithm with constant performance guarantee when we use the term *approximation algorithm.* On the other hand, there are problems for which for each $k \in (1, \infty)$ there is an approximation algorithm with performance guarantee k. In a manner of speaking, the problem is arbitrarily well approximable. This leads us to polynomial time approximation schemes.

Definition 1.11 (polynomial time approximation scheme). *Let $P = (\mathcal{I}, S, c, \mathrm{opt})$ be an optimization problem. An algorithm is called a* polynomial time approximation scheme *(*PTAS *for short) if for each instance $I \in \mathcal{I}$ and each $k > 1$ it computes a solution $s \in S(I)$ with performance ratio $R_P(I, s) \leq k$ in polynomial time.*

A disadvantage of a PTAS can be the increase of the running time when we want to achieve better and better approximations. To be more precise, if the performance ratio is k, the running time might not be polynomial in $1/(k-1)$.

Complexity Classes. Problems are assigned to *complexity classes*, depending on the asymptotic behaviour of the time complexity function of the algorithms for these problems. Literally, there is a proper *zoo* of complexity classes, cf. [27], where about 500 complexity classes are listed. At this point, we define only the classes which are used in this thesis.

- $\mathcal{P}$: The class of all decision problems P which are solvable in polynomial time, i.e. for which there exists a polynomial f and an algorithm that solves P correctly and whose time complexity function is in $\mathcal{O}(f)$.
- $\mathcal{PO}$: The same as $\mathcal{P}$ for optimization problems.
- $\mathcal{NP}$: The class of all decision problems for which for each instance whose answer is "YES", there is a proof for the correctness of this anwer, and this proof can be verified in polynomial time.
- $\mathcal{NPO}$: The class of all optimization problems for which the corresponding decision problem, as described in Remark 1.8, is in $\mathcal{NP}$.

$\mathcal{APX}$: The class of all problems in $\mathcal{NPO}$ for which there is an approximation algorithm.

$\mathcal{PTAS}$: The class of all problems in $\mathcal{NPO}$ for which there is a PTAS.

We omit a stringent definition of a subclass of $\mathcal{APX}$, namely the class max $\mathcal{SNP}$, at this point because we will only mention it in one paragraph but not investigate it more closely. In addition, the class max $\mathcal{SNP}$ is defined in a syntactical way. In our context, this seems rather unhandy in contrast to the classes above, which are defined from a computational perspective.

Reductions and Completeness. In this thesis, we prove several problems as *complete*. Therefore, we reduce problems to each other. In this paragraph, we define three different types of reductions, specify what completeness is and how it can be derived by reducing problems, and explain why the concept of completeness is useful for the development of algorithms. The notions of $\mathcal{NP}$-completeness and polynomial time reduction can be found in every text book on theoretical computer science. The definitions for AP- and L-reduction are borrowed from [30] and adapted to our notations. This reference gives also a nice overview of several other types of reductions.

Definition 1.12 (polynomial time reduction, $\mathcal{NP}$-hard, $\mathcal{NP}$-complete). *A* polynomial time reduction *from a decision problem $P = (\mathcal{I}_P, S_P) \in \mathcal{NP}$ to a decision problem $Q = (\mathcal{I}_Q, S_Q) \in \mathcal{NP}$ is a function $f : \mathcal{I}_P \to \mathcal{I}_Q$ with*

- $S_P(I_P) = \text{“YES”} \iff S_Q(f(I_P)) = \text{“YES”}$, *and*
- $f \in \mathcal{O}(p)$ *for some polynomial p.*

Computable functions with the second property are said to be polynomial time computable. *If there is such a reduction from P to Q, then this is denoted by $P \leq Q$.*

*A problem Q is $\mathcal{NP}$-*hard *if $P \leq Q$ holds for all $P \in \mathcal{NP}$. It is $\mathcal{NP}$-*complete *if additionally $Q \in \mathcal{NP}$.*

Polynomial time reductions as defined above are also called *many-one reductions* because the function f is not required to be injective. The importance of $\mathcal{NP}$-completeness is the possibility to derive a polynomial time algorithm for *every* problem in $\mathcal{NP}$ from a hypothetical polynomial time algorithm for one $\mathcal{NP}$-complete problem. In this sense, an $\mathcal{NP}$-complete problem can be seen as a hardest one in the class $\mathcal{NP}$. It is well-known that there is no polynomial time algorithm for an $\mathcal{NP}$-complete problem, unless $\mathcal{P} = \mathcal{NP}$.

The other notions of reducibility concern optimization problems. At first, we describe *AP-reductions*, a type of reduction which preserves membership in $\mathcal{APX}$. An *L-reduction* even preserves membership in $\mathcal{PTAS}$, thus it is a more specific reduction. However, if the problem which shall be reduced is a minimization problem, then L-reducibility implies

AP-reducibility. That is the statement of Lemma 1.15. Since an L-reduction is often easier to construct, one usually prefers this kind of reduction in this case.

To get a rough idea of both types of reduction, consider the diagram below.

$$\begin{array}{ccc} I_P & \xrightarrow{\quad f \quad} & I_Q \\ {\scriptstyle \text{PTAS for } P}\big\downarrow & & \big\downarrow{\scriptstyle \text{PTAS for } Q} \\ S_P(I_P) \ni g(I_P, s_Q) = s_P & \xleftarrow{\quad g \quad} & s_Q \in S_Q(f(I_P)) \end{array}$$

The function f transforms an instance I_P of the problem P to an instance $I_Q = f(I_P)$ of problem Q. Now assume that there was a polynomial time approximation scheme which computes a solution $s_Q \in S_Q(f(I_P))$ for Q. Then the function g could be used to compute a solution $s_P \in S_P(I_P)$ for I_P from this transformed instance and its solution. As apparent in the diagram, a PTAS for problem P could be composed by concatenating the function f, the PTAS for Q, and the function g. We remark that it is sufficient to define g only for instances on Q which actually arise as transformation of f. This is why we chose $\mathcal{I}_P$ as the first part of the domain of g instead of $\mathcal{I}_Q$.

The preservation of approximability is treated differently for both types of reduction. For an AP-reduction, the existence of a PTAS for problem Q would enable us to construct a PTAS for problem P. In the case of L-reductions, the approximability of the concatenated operation is demanded explicitly. Moreover, also the function f is required to preserve approximability. We proceed with the formal definitions of both reductions.

Definition 1.13 (AP-reduction, $\mathcal{APX}$-hard, $\mathcal{APX}$-complete). *An* AP-reduction *from an optimization problem $P = (\mathcal{I}_P, S_P, c_P, \mathrm{opt}_P)$ to another optimization problem $Q = (\mathcal{I}_Q, S_Q, c_Q, \mathrm{opt}_Q)$ is a triple (f, g, α_{AP}), where $f : \mathcal{I}_P \times \mathbb{Q}_{>1} \to \mathcal{I}_Q$ and $g : \mathcal{I}_P \times S_Q \times \mathbb{Q}_{>1} \to S_P$ are polynomial time computable functions, α_{AP} a positive constant, and the following properties hold for all $I_P \in \mathcal{I}_P$, $r \in \mathbb{Q}_{>1}$, and for all $s_Q \in S_Q(I_Q)$ with $I_Q = f(I_P, r) \in \mathcal{I}_Q$:*

- $S_P(I_P) \neq \emptyset \Rightarrow S_Q(I_Q) \neq \emptyset$,
- $g(I_P, s_Q, r) \in S_P(I_P)$,
- $R_Q(I_Q, s_Q) \leq r \Rightarrow R_P(I_P, g(I_P, s_Q, r)) \leq 1 + \alpha_{AP}(r - 1)$.

*This reduction is denoted by $\leq_{AP}$. Similarly as above, a problem Q is $\mathcal{APX}$-*hard *if $P \leq_{AP} Q$ holds for all $P \in \mathcal{APX}$, and $\mathcal{APX}$-*complete *if in addition Q is in $\mathcal{APX}$.*

If an optimization problem is $\mathcal{APX}$-complete then there is no PTAS for it unless $\mathcal{P} = \mathcal{NP}$. The letters "AP" stand for "approximation preserving".

Definition 1.14 (L-reduction). *An* L-reduction *from an optimization problem $P = (\mathcal{I}_P, S_P, c_P, \mathrm{opt}_P)$ to an optimization problem $Q = (\mathcal{I}_Q, S_Q, c_Q, \mathrm{opt}_Q)$ is a 4-tuple (f, g, α, β),* *where* $f : \mathcal{I}_P \to \mathcal{I}_Q$ *and* $g : \mathcal{I}_P \times S_Q \to S_P$ *are polynomial time computable functions,* α *and* β *are positive constants, and the following properties hold for all* $I_P \in \mathcal{I}_P$ *and for all* $s_Q \in S_Q(I_Q)$ *with* $I_Q = f(I_P) \in \mathcal{I}_Q$*:*

- $S_P(I_P) \neq \emptyset \Rightarrow S_Q(I_Q) \neq \emptyset$,
- $g(I_P, s_Q) \in S_P(I_P)$,
- $\mathrm{opt}_Q(I_Q) \leq \alpha \cdot \mathrm{opt}_P(I_P)$,
- $|\mathrm{opt}_P(I_P) - c_P(I_P, g(I_P, s_Q))| \leq \beta \cdot |\mathrm{opt}_Q(I_Q) - c_Q(I_Q, s_Q)|$.

Similarly to the notation above, we denote $P \leq_L Q$ *if there is an L-reduction from* P *to* Q*.*

Here, "L" usually stands for "linear" but there are also references which use it for "logical-first order" ([8]). We remark that also the terms $\mathcal{PTAS}$*-completeness* and max $\mathcal{SNP}$*-completeness* can be defined, which are related to L-reductions. Getting back to problems in $\mathcal{APX}$, it is known that every max $\mathcal{SNP}$-complete problem is also $\mathcal{APX}$-complete ([4]).

In Chapter 2 we will prove a maximization problem as $\mathcal{APX}$-complete by reducing an $\mathcal{APX}$-complete minimization problem to it. The lemma below states that this can be done via an L-reduction. We remark that a similar proof, though of another statement, can be found in [29].

Lemma 1.15 (Special case of Lemma 8.2 in [7]). *Let* P *be a minimization problem,* Q *a maximization problem, and* $P \leq_L Q$*. Then* P *also reduces to* Q *via an AP-reduction, i.e.* $P \leq_{AP} Q$*.*

Proof. Because P is a minimization and Q a maximization problem, the last item in Definition 1.14 reads as $c_P(I_P, g(I_P, s_Q)) - \min(I_P) \leq \beta(\max(I_Q) - c_Q(I_Q, s_Q))$. Dividing this by the inequality in the third item one gets

$$\frac{c_P(I_P, g(I_P, s_Q)) - \min(I_P)}{\min(I_P)} \leq \alpha\beta \frac{\max(I_Q) - c_Q(I_Q, s_Q)}{\max(I_Q)}. \tag{1.9}$$

On the other hand, from $0 \leq (\max(I_Q) - c_Q(I_Q, s_Q))^2$ we derive

$$\frac{\max(I_Q) - c_Q(I_Q, s_Q)}{\max(I_Q)} \leq \frac{\max(I_Q)}{c_Q(I_Q, s_Q)} - 1. \tag{1.10}$$

Plugging (1.10) into (1.9) we obtain

$$\frac{c_P(I_P, g(I_P, s_Q))}{\min(I_P)} - 1 \leq \alpha\beta \left(\frac{\max(I_Q)}{c_Q(I_Q, s_Q)} - 1 \right). \tag{1.11}$$

Setting $\alpha_{AP} = \alpha\beta$, the third item in Definition 1.13 can directly be derived from (1.11), namely that $\max(I_Q)/c_Q(I_Q, s_Q) \leq r$ implies $c_P(I_P, g(I_P, s_Q))/\min(I_P) \leq 1 + \alpha_{AP}(r-1)$. The functions g in both definitions can be identified, where the variable r, which appears in function g in Definition 1.13, may be ignored. The same can be done with function f. $\square$

Chapter 2

Spanning Trees with Many Leaves

This chapter is dedicated to the problem of finding a spanning tree of a graph with as many leaves as possible. We start with a short historical outline, followed by an overview of results concerning the complexity, the (non-)approximability, and of exact algorithms. In Section 2.2, we give three areas of applications where it is crucial to search for trees with many leaves. Section 2.3 contains the proof of $\mathcal{NP}$-completeness of the MLST for cubic planar graphs and the further restriction to graphs which are additionally biconnected. For regular graphs, the $\mathcal{APX}$-completeness of the problem is the matter of Section 2.4. We also discovered—admittedly very restrictive—classes of graphs for which there are polynomial time algorithms for the computation of spanning trees with the maximum number of leaves. This is given in Section 2.5. Parts of this chapter are submitted, see [107].

Contribution. Garey and Johnson ([53]) established the $\mathcal{NP}$-completeness of the MAXIMUM LEAF SPANNING TREE Problem for general graphs. Lemke ([73]) went one step further and showed that the problem is $\mathcal{NP}$-complete even for cubic graphs. In Section 2.3, Lemke's proof is modified such that the result does also hold for cubic graphs which are in addition planar and 2-connected.

In [20] and [28] the authors conjectured max $\mathcal{SNP}$-hardness of the MLST for cubic graphs. The $\mathcal{APX}$-completeness for cubic graphs then was shown in [19]. We provide $\mathcal{APX}$-completeness proofs for k-regular graphs for each odd $k \geq 5$.

After a little observation we list up several classes of graphs, for which the MAXIMUM LEAF SPANNING TREE Problem is solvable in polynomial time, or admits a PTAS at least.

2.1 Introduction

A basic task in graph theory is to find a spanning tree for a given graph that satisfies certain conditions. The field of such tree spanner problems is a *very* broad one—thus, we omit an overview at this point. The issue of this chapter is the problem to find a spanning tree that maximizes the number of leaves over all spanning trees, i.e. a *maximum leaf spanning tree*, or MLST for short. The *maximum leaf number* is the number of leaves of an MLST. The problem itself is the MAXIMUM LEAF SPANNING TREE Problem (MLST).

From a historical point of view, the MLST seems not to be considered much earlier than 1979, what primarily is due to its relationship to the MINIMUM CONNECTED DOMINATING SET Problem (MCDS). The objective of this problem is to find a *minimum connected dominating set* (MCD) in a graph $G = (V, E)$, i.e. a minimum subset $S \subseteq V$ for which $G[S]$ is connected and every vertex in $V \setminus S$ is adjacent to a vertex in S (cf. GT2 in [53]). Computing now an arbitrary spanning tree on $G[S]$ and join each vertex in $V \setminus S$ to this tree results in a maximum leaf spanning tree for G. In this way, the MCDS and the MLST are equivalent. The more general problem of looking for a dominating set which is not necessarily connected can firstly be found in a 1962 book from Ore ([97]). Connected dominating sets were initially investigated in 1979, after "Hedetniemi suggested a new parameter in domination theory", see [111]. In the same year, the famous book of Garey and Johnson ([53]) was published, in which the MLST appears as Problem ND2.

As stated there, the problem is $\mathcal{NP}$-complete for general graphs. This is also true for a range of special classes of graphs. Among these classes are planar graphs with maximum degree 4, 4-regular graphs (both [53]), as well as cubic graphs ([73]). We extend this list by showing the $\mathcal{NP}$-completeness of the problem restricted to graphs which are planar *and* cubic.

Considering the approximability of the MLST, it turns out to be max $\mathcal{SNP}$-hard on general graphs (Galbiati et al. [51]). Hence, the approximability has been studied exhaustively in recent years. Particularly, Lu and Ravi gave a 3-approximation for general graphs in [85] and Solis-Oba provided a 2-approximation for general graphs in [118]. Loryś and Zwoźniak studied the problem restricted to cubic graphs and presented a 7/4-approximation in [84]. This factor was improved to 5/3 by Correa et al. in [28] and to 3/2 by Bonsma and Zickfeld in [20]. When taking a closer look at the max $\mathcal{SNP}$-hardness proof in [51], it becomes clear that it can be adapted to graphs with bounded degree. We go a step further and show that MLST is $\mathcal{APX}$-complete even on 5-regular graphs. A slightly different result—the max $\mathcal{SNP}$-completeness for cubic graphs—had been conjectured in [20] and in [28]. With our result, it could have been possible to show the max $\mathcal{SNP}$-completeness or the $\mathcal{APX}$-completeness of the MLST for cubic graphs. However, this was recently shown by Bonsma in [19] by using a different approach.

In another line of research exact exponential algorithms for the MLST were investigated. The first one seems to be released in [50] where the author also reported on computational results on random and on grid graphs. However, he did not specify the running time of his algorithm. A further exact algorithm with running time $\mathcal{O}(1.9407^n)$ was presented in [46]. In [44], this running time could be improved to $\mathcal{O}(1.8966^n)$. Very recently, exact algorithms for the directed case were studied. In doing so, running times of $\mathcal{O}^*(1.9044^n)$

and $\mathcal{O}^*(1.8139^n)$, respectively, could be achieved, where the latter one required exponential space, see [15].

There are several results concerning lower bounds on the maximum leaf number for cubic and other regular graphs, as well as for graphs with different minimum degrees greater than 2. All values range in $\Theta(n)$. For more detailed overviews see e.g. the introductions in [19, 28, 84]. Exact values for special grid graphs were provided in [75].

2.2 Applications

The Maximum Leaf Spanning Tree Problem is not only interesting from an theoretical point of view. In this section, we describe three practical applications where trees with many leaves turn out to be useful. All three descriptions are based on the cited references, therein.

Rendering of Triangulated Meshes. An important topic in computer graphics is *rendering*. More detailed, a three-dimensional object can be described by just its surface, which can be a triangulated mesh, embedded into the three-dimensional Euclidean space. During the rendering process, one triangle after the other is projected onto the view plane, e.g. the screen.

The access to the triangles is realized by their coordinates, which have to be transferred from the memory to the arithmetic logic unit. This, in turn, is done by using fast cache memories. But these cache memories work efficiently only if data from the cache is repeatedly accessed, so one tries to keep vertices in the cache.

Now reconsider the triangulated mesh. A *triangle strip* is a sequence of triangles of this mesh, where consecutive triangles have one *shared edge* in common, and in which no triangle appears twice. The access to the triangles is carried out faster if the triangles are treated in the order of such a strip since two of the three vertices are already in the cache due to the preceding triangle.

Finding such a strip which contains all triangles is equivalent to find a Hamiltonian path in the dual graph, and hence is $\mathcal{NP}$-complete. Alternatively, one can use more than one strip, or one adds a small number of triangles to the graph in order to be able to find a single strip.

Now consider those edges which are not shared edges. These edges form a spanning tree if the graph has a positive genus. It emerged that cache reuse is higher for vertices that are leaves of this spanning tree, since each of these leaf vertices is accessed several times successively. Thus, one is interested in the Mlst.

For a more elaborate description of this application we refer to [34]. To compute a spanning tree with many leaves, the authors avoided to implement known algorithms like the 3-approximation of Lu and Ravi, which is however mentioned there. Instead, they report that “A breath first spanning tree with low depth and large fan-out would maximize the number of leaf vertices ...”.

Design of Wireless Ad-hoc Networks. A wireless ad-hoc network can be seen as a set of devices which communicate to each other. As of now, the devices are referred to as nodes. All of these nodes are uniform, hence, there is no node which serves as a basis station or something similar. Two nodes can communicate directly to each other using a single hop or by using multihops via other nodes in the network. The nodes can be able to move and thus get out of range of each other. Hence, such a network is organized dynamically.

Originally, wireless ad-hoc networks seem to have appeared in military applications for the first time ([11]). Of course, there are also many non-military applications, which include for example communication between mobile devices like notebooks or cell phones.

To describe how the MLST can be applied to wireless ad-hoc networks, it is more convenient to switch to the equivalent MINIMUM CONNECTED DOMINATING SET Problem. Since it is more involved to communicate with multihops, one seeks for a *small* set of nodes which serve as transmitting nodes. This set is demanded to be *connected* in order that the transmitting nodes are able to communicate to each other. And finally, it must be *dominating* so that also all other nodes can be reached.

For an overview of the research on the MCDS in the context of wireless networks we refer again to [11], which also provides the application of coloring and clique problems to the area of communication networks.

Regenerator Location in Optical Networks. In an optical network, pulses of light are sent through optical fibers to transmit information. It is possible to use many different frequencies which increases the capacities of the fibers. Optical signals fall off in intensity and in quality in the course of the transmitting process. Thus, they are required to be regenerated periodically after a maximum distance.

More precisely, the signals can be reamplified, reshaped, and retimed. For the mathematical model it is more convenient to use only regenerators which perform all three forms of regeneration. Moreover, “the associated equipment is usually expensive and it [the placement of the regenerators] tends to be done at nodes of a telecommunication network.”, see [26]. Furthermore, the signal’s intensity does not decrease continuously in the model, rather it is simply lost after it had been covered the maximum distance. As one might expect, one is interested in a possibly small set of regenerators in an optical network, such that signals can be transmitted between all pairs of nodes.

This leads to the REGENERATOR LOCATION Problem (RL), which is investigated more detailed for example in [26]. The authors of this paper also provide heuristic algorithms for this problem and give computational results for their heuristics. Formally defined, the decision version of this problem reads as follows.

REGENERATOR LOCATION		
Instance:	Graph	$G = (V, E)$,
	distance function	$d : E \to \mathbb{R}_{\geq 0}$,
	maximum distance	$d_{\max}$,
	integer	k.
Question:	Is there a subset	$R \subseteq V$, $\lvert R\rvert \leq k$
	such that	$\forall u, v \in V \;\; \exists P \in \mathcal{P}_{u,v}$
	with	$\mathrm{dist}_G(u, v) > d_{\max} \Longrightarrow \exists x \in P \cup R$?

We define the *communication graph* $M = (V, E(M))$ with $E(M) := \{\{u, v\} \mid \mathrm{dist}_G(u, v) \leq d_{\max}\}$ and $d(uv) := \mathrm{dist}_G(u, v)$ for all $uv \in E(M)$ for a graph $G = (V, E)$ and a maximum distance $d_{\max}$, in which we follow [26]. Then it turns out that a solution of the RL Problem on the communication graph is also a solution of the RL Problem on the original graph. Now, Lemma 2 in [26] states that "Any minimal solution for the RL Problem on M can be represented as a spanning tree with regenerators at all internal nodes of the tree.", thus, it is nothing else than the MAXIMUM LEAF SPANNING TREE Problem on M. This however is only true if the communication graph is not complete. In this case, $R = \emptyset$, but the star tree has one internal node.

2.3 $\mathcal{NP}$-completeness of the MLST on Planar Cubic Graphs

In this section, we establish the $\mathcal{NP}$-completeness of the MAXIMUM LEAF SPANNING TREE Problem on graphs that are both, planar and cubic. The proof will be by reduction of a planar version of the EXACT COVER BY 3-SETS Problem (SP2 in [53]), where we will benefit from the embedding of the corresponding bipartite graph and from a special way to connect the 3-sets. This construction is described in Subsection 2.3.1 in a general manner. In Subsection 2.3.2, the problems used in this section are defined. Subsection 2.3.3 is firstly dedicated to specify the general construction of connecting the 3-sets described in Subsection 2.3.1 to our reduction. Afterwards, we describe the design of the required gadgets for the reduction. The section is closed with the $\mathcal{NP}$-completeness result of the MLST Problem on planar cubic graphs. We remark that stringent definitions for planar graphs and related notions like *face* and *boundary* are given in Subsection 3.4.1. In this context, we will also define the dual of a planar graph and describe a relationship between strictly fundamental cycle bases and the OPTIMUM COMMUNICATION SPANNING TREE Problem on planar graphs.

2.3.1 Connection of the 3-Sets

Consider a problem whose instances contain amongst others two or more disjoint sets. Besides Exact Cover by 3-Sets (X3C), such problems are for example 3-Dimensional Matching or several variants of Satisfiability (Sat). When reducing instances of these problems to prove the completeness of decision or optimization problems, we observe a general technique, in particular, when the decision respectively the optimization problem is a spanning tree problem.

X3C and Sat are designed in a fashion that an element of one of the disjoint sets *contains* (in some way) elements of the other set in the instance, e.g. a 3-set contains elements of the other set (X3C), and a clause contains literals (Sat). The elements of the sets are modeled as vertices, and if an element of one set contains an element of the other set, the corresponding vertices are joined to each other. We refer to nodes which correspond to sets as *set nodes* and to the other ones as *element nodes* in this section and in Section 3.5. Observe that graphs derived in this way are bipartite. Now, this bipartite graph is equipped with some further structure which connects the set nodes to each other. Examples for this kind of reduction can be found in [62] (X3C $\leq$ Shortest Total Path Length Spanning Tree, ND3 in [53]) and [73] (X3C $\leq$ Maximum Leaf Spanning Tree on cubic graphs) or, more recently, in [91] (3DM $\leq$ Minimum Vertex Ranking Spanning Tree) and in [52] (Max-3Sat-Nae-Un-q $\leq_L$ Msfcb). For the definitions of these problems we refer to the quoted references.

The problems 3Sat, X3C and 3DM remain $\mathcal{NP}$-complete when the bipartite graph described in the paragraph above is planar, see [41,76]. But in general, the supplementary structure for the connection of the 3-sets destroys planarity. Thus, it is not possible to transfer a completeness result for a problem on general graphs to the problem on planar graphs in a straightforward manner.

Our approach is to connect the set nodes with edges and nodes passing through the faces of the embedded graph. This is done in a way that for each pair of set nodes, there is exactly one path between them which uses only these new edges. Restricted to the set nodes, the new edges thus form a spanning tree, which we call the *spinal tree*. Given that the bipartite graph is simple—which might be realized by an additional pre-processing step—the boundary of each face contains at least two set nodes. Thus, such a construction can always be realized while preserving planarity. Moreover, the transformation can be constructed in polynomial time if the size of the supplementary structure added into the faces is polynomially bounded.

We will apply these ideas both in this section to prove the $\mathcal{NP}$-completeness of the Mlst on planar cubic graphs and in Chapter 3 which deals with strictly fundamental cycle bases and an $\mathcal{NP}$-completeness result for their minimization restricted to planar graphs.

2.3.2 The Problems

This subsection is devoted to fix the notations of the discussed problems. Firstly, we define the Maximum Leaf Spanning Tree Problem restricted to planar cubic graphs.

3-P-Mlst	
Instance:	Planar cubic graph $G = (V, E)$, positive integer $k \leq \lvert V \rvert$.
Question:	Does G have a spanning tree with at least k leaves?

Similarly to [73], we prove the $\mathcal{NP}$-completeness of the more specific question about the existence of a spanning tree T on a given planar and cubic graph G_O without a vertex v with $\deg_T(v) = 2$. We call this the Planar Odd Degree Spanning Tree Problem, 3-P-Odst for short, when the problem is restricted to cubic graphs. ODST denotes the *odd degree spanning tree* itself. Actually, the 3-P-Odst is the special case of the 3-P-Mlst with $k = \frac{n}{2} + 1$, since if there is an odd degree spanning tree in a cubic graph, it has $\frac{n}{2} - 1$ vertices of degree 3 and $\frac{n}{2} + 1$ leaves. This is also true for cubic graphs which are not necessarily planar. Our proof will use reduction from the planar version of Exact Cover By 3-Sets, whose $\mathcal{NP}$-completeness had been established in [41]. It is defined as follows.

P-X3C	
Instance:	Integers n, m with $3 \vert n$, set $X = \{1, 2, \ldots, n\}$, subset $\mathscr{S} \subset \mathcal{P}(X)$ with $\lvert \mathscr{S} \rvert = m$ and $\lvert S \rvert = 3$ for all $S \in \mathscr{S}$, where $G_X = (V_X, E_X)$ with $V_X = X \cup \mathscr{S}$ and $E_X = \bigcup_{S \in \mathscr{S}} \{\{S, x\} \mid x \in S\}$ is planar.
Question:	Is there a subset $\mathscr{S}' \subseteq \mathscr{S}$ such that for all $x \in X$ there is exactly one $S \in \mathscr{S}'$ with $x \in S$?

2.3.3 The Transformation

Initially, consider a P-X3C instance with the associated graph G_X and fix a plane embedding of G_X. Note that in general this embedding is not unique. However, this does not affect the proof. This embedding will be transformed into a planar and cubic graph G_O for the 3-P-Odst instance. We will show that the graph G_O has a spanning tree with no vertices of degree 2 iff there is a subset $\mathscr{S}' \subseteq \mathscr{S}$ as posed in the description of P-X3C in Subsection 2.3.2.

We start the transformation with the construction of the spinal tree. In detail, for each set node $S \in \mathscr{S}$ put a *facial node* between each of the three pairs of element nodes adjacent to S and connect these facial nodes with *join edges* to S, see Figure 2.1 for this setting. Now merge some of the facial nodes together in such a way that for each pair in $\mathscr{S}$ there is a unique path which consists only of join edges. Note that merged nodes have to be located in the same face. Figures 2.1 and 2.2 illustrate this construction with an example with

$X = \{1, 2, \dots, 6\}$ and $\mathscr{S} = \{\{2, 3, 5\}, \{1, 2, 6\}, \{1, 4, 6\}, \{1, 5, 6\}\}$. The element nodes are depicted as numbered circles, while the rectangles represent the set nodes.

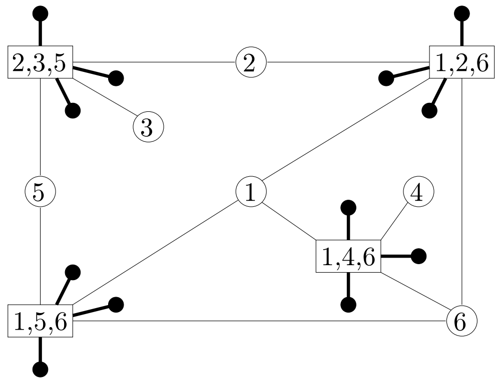


Figure 2.1: G_X with additional facial nodes (filled bullets) and join edges (fat lines).

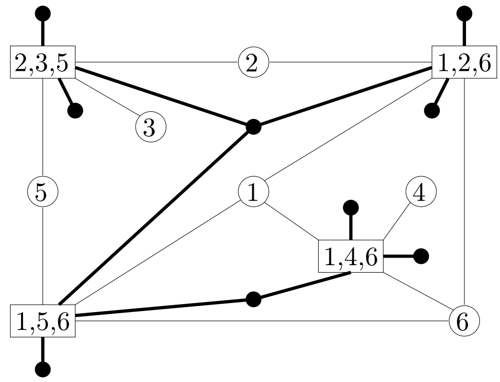


Figure 2.2: G_X after merging several facial nodes.

Since the boundary of every face contains at least two nodes from $\mathscr{S}$, this construction can always be done while preserving planarity. For the further transformation to G_O we replace the nodes and edges with components which we describe below.

Plugs. This component is required to substitute element nodes and facial nodes with degree 1. In both cases, the incident edge is identified with edge e in Figure 2.3, which shows the only two possible ODSTs restricted to a plug. In these cases, the spanning tree is called *regular*[1] restricted to the plug, otherwise *irregular* on the plug. Note that the edge e has to be in every spanning tree since it is a bridge.

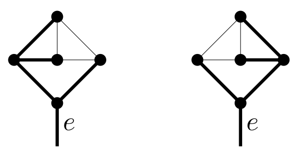


Figure 2.3: Two possibilities of a regular spanning tree on a plug.

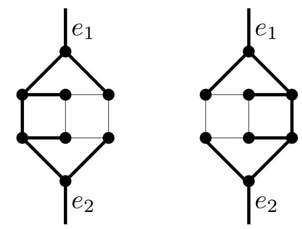


Figure 2.4: Two possibilities of a regular spanning tree on a connector.

Connectors. A connector replaces the element nodes and facial nodes with degree 2. Furthermore, connectors are required for the construction of a gear, which is defined later.

[1]This should not be confused with the 3-regularity of the graph G_O itself.

The two incident edges are identified with e_1 and e_2 in Figure 2.4. Again, there are only two possible ODSTs restricted to a connector, and both contain e_1 and e_2. Also when considering the whole graph, e_1 and e_2 are connected in the tree only via edges of the connector, otherwise, the spanning tree would have nodes of degree 2 in this connector. In this case, it is referred to as *irregular* on this connector. The two depicted trees are thus *regular* on the connector.

Lanes. With a lane we replace element nodes and facial nodes with degree 3 or more. For such a node with degree j we construct a row of $j-2$ houses as displayed in Figure 2.5. The lane subgraph has also been constructed by Lemke in [73]. Note that it has j outgoing edges.

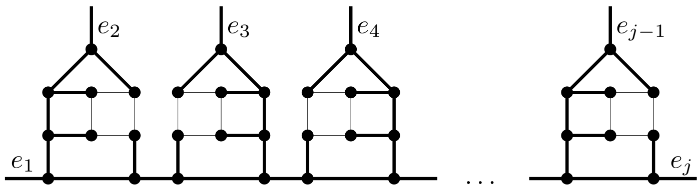


Figure 2.5: A lane with a possible ODST.

Figure 2.5 shows one of the 2^{j-2} possible odd degree spanning trees restricted to this subgraph (the top of every house is connected to the bottom either on the left or on the right hand side). Observe that all edges $e_1, \dots, e_j$ are in every ODST and that they are pairwise reachable to each other on tree edges only in the lane. Again, if all degrees of a spanning tree on a lane are odd, the tree is *regular* on the lane and *irregular* otherwise.

Switchers. Switchers will be pieces of the gears introduced in the next paragraph. See Figure 2.6 for a switcher. The grey edges will be outgoing edges of the components defined above. Hence, they are forced into any odd degree spanning tree by these components. Figure 2.6 displays the only two possibilities of assigning the remaining edges to an ODST. Once again, if all degrees of a spanning tree restricted to a switcher are odd, it is called *regular* on the switcher and *irregular* otherwise. However, in the case of a switcher, the two possibilities of a regular spanning tree essentially differ from each other.

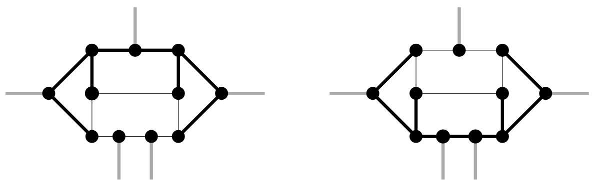


Figure 2.6: Spanning tree which is inactive (left) and active (right) on a switcher.

In every ODST the grey edge on the ride hand side is reachable from the left one by using only tree edges contained in the switcher. As the illustration shows, either the upper grey edge or the two lower grey edges are connected to the switcher via tree edges. The first possibility we call *inactive*, the second one we refer to as *active*.

Gears. Now we are able to construct the component for the set nodes, which we call a *gear*. This subgraph consists of three inner connectors, three outer connectors, and three switchers. All pieces are assembled as depicted in Figure 2.7. The edges f_1, f_2 and f_3 belong to components that replaced facial nodes, the edges x_1, x_2 and x_3 are parts of components that replaced the nodes in X. With the assumption that the grey edges are forced into the tree by other components, the fat edges constitute one of the 2^6 possibilities for an odd degree spanning tree restricted to the gear without considering the switchers.

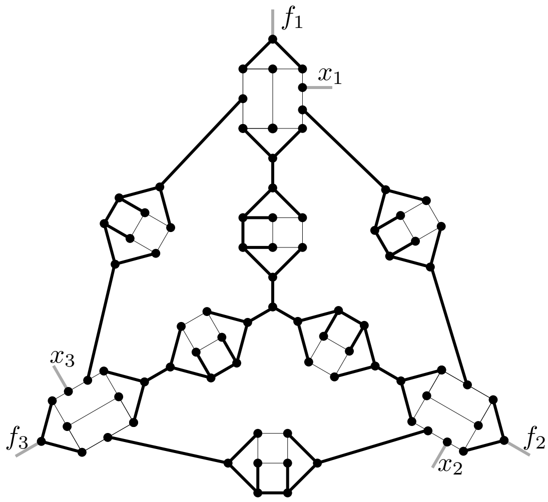


Figure 2.7: A gear with a partial spanning tree.

Provided that the grey edges are in the tree and that the six connectors as well as the three switchers are regular, we observe that all three switchers in a gear have to be either active or inactive to preserve connectivity and to avoid circuits in the spanning tree. Thus, the terms *regular*, *irregular*, *active*, and *inactive* can be passed on to gears. Active gears correspond to 3-sets $S \in \mathscr{S}'$, inactive ones to sets $S \in \mathscr{S} \setminus \mathscr{S}'$.

Observation 2.1. *The transformation can be performed in polynomial time. Moreover, the constructed graph is planar and cubic.*

Lemma 2.2. *Let $\mathscr{S}$ and X belong to an instance of* P-X3C *and G_O be a graph transformed in the way described above. Then G_O has an ODST T if and only if there exists a subset $\mathscr{S}' \subseteq \mathscr{S}$ such that for all $x \in X$ there is exactly one $S \in \mathscr{S}'$ with $x \in S$.*

Proof. Observe at first that a tree T on G_O can only be an ODST if

- all plugs, connectors, lanes, and gears are regular,

- all gears are connected to each other in T via join edges and the replacements of facial nodes, and
- every element in $x \in X$ is connected directly to all the gears which represent sets $S \in \mathscr{S}$ with $x \in S$, where *directly* means not by using other nodes in X or nodes of other gears.

Assume that the given instance of P-X3C has a solution $\mathscr{S}' \subseteq \mathscr{S}$. Then, construct for every set in $\mathscr{S}'$ an active gear, and for the sets in $\mathscr{S} \setminus \mathscr{S}'$ an inactive gear.

It remains to show that the chosen edges form a spanning tree. This is fulfilled since every $x \in X$ is connected to exactly one active gear and thus, disconnectivity and cycles are avoided.

The other way around, suppose now that there is no solution of the P-X3C instance. Then for every choice $\mathscr{S}' \subseteq \mathscr{S}$ there is an $x \in X$ with

$$|\{S \in \mathscr{S}' \mid x \in S\}| = 0 \text{ or with} \tag{2.1}$$

$$|\{S \in \mathscr{S}' \mid x \in S\}| \geq 2. \tag{2.2}$$

With the observations at the begin of this proof in mind, we can construct a cubic subgraph on G_O. But this subgraph cannot be a tree since there is an x which is not connected to the rest of the subgraph (2.1) or it induces a circuit since it is connected to at least two gears (2.2), which are also connected via the spinal tree. □

As one can easily see, the problems 3-P-ODST and 3-P-MLST are in $\mathcal{NP}$. Thus, Lemma 2.2 together with Observation 2.1 leads to

Theorem 2.3. ODST *is $\mathcal{NP}$-complete on planar cubic graphs.* □

Since the 3-P-ODST is a special case of the 3-P-MLST we conclude like in [73]

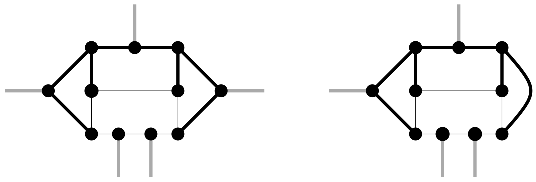
Theorem 2.4. *The* MAXIMUM LEAF SPANNING TREE *Problem is $\mathcal{NP}$-complete on planar cubic graphs.* □

2.3.4 Restriction to Biconnected Graphs

The statement of Theorem 2.4 does also hold if the planar and cubic graph is additionally required to be 2-connected. In this subsection, we show how the reduction must be modified to obtain this result.

Our aim is to avoid the vertices of degree 1 after the merging process of the facial nodes, see again Figure 2.2. There are two types of degree 1 vertices, element nodes and facial nodes, and both types are treated differently. Anyway, the usage of the plugs becomes obsolete.

Facial Nodes. Firstly consider the facial nodes which are not merged. For the desired connection of the set nodes they are useless, and thus, we simply remove them. But then, the switcher from which the corresponding plug was removed, also has to be adapted. See Figure 2.8 for the modification.



(a) (b)

Figure 2.8: An inactive switcher (a). The switcher after the modification (b).

The only two possibilities of an odd degree spanning tree which contains the grey edges can directly be carried over from the original switcher.

Element Nodes. For the element nodes which are contained in exactly one set in $\mathscr{S}$, we perform a preprocessing step on the embedded instance of the P-X3C. Contrary to facial nodes, element nodes cannot simply be removed. Recall that the graph is simple and bipartite and hence, each face boundary contains at least two element nodes, especially these faces which contain an element node of degree 1. Therefore, a further gadget can be attached into the face between a degree 1 element node and one of the at least two element nodes without destroying the planarity. See Figure 2.9 for the gadget and its insertion.

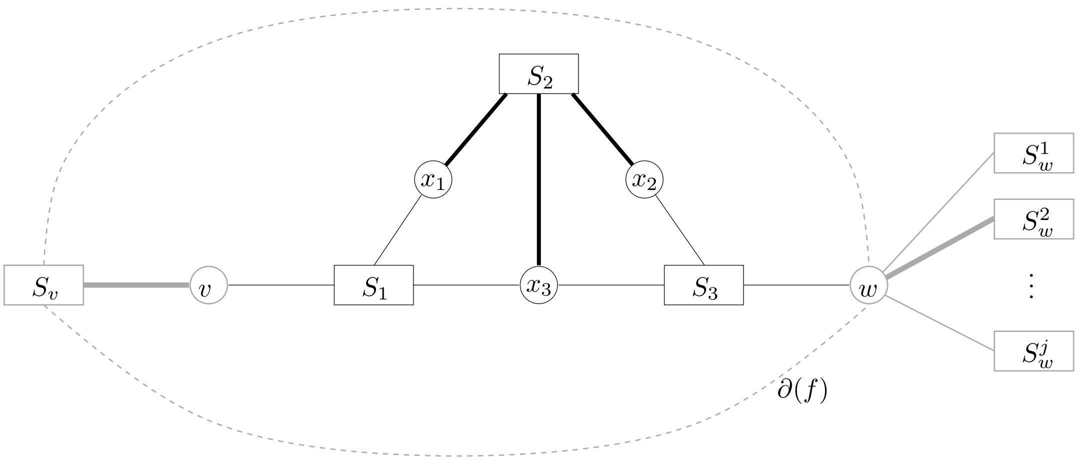


Figure 2.9: Gadget to avoid element nodes with degree 1. The original partial instance is drawn in grey, the inserted gadget in black.

In Figure 2.9, there is an element node v with degree 1 in the original instance. This node lies in a face f whose boundary $\partial(f)$ contains an element node w. Let this node w be an element in j subsets $S_w^1, \ldots, S_w^j$. Assume at first that the original instance has a solution. Then, v must be covered by S_v and w by one of the j subsets. Figure 2.9 indicates the only possibility how the new element nodes x_1, x_2 and x_3 can be covered. Now suppose that there is no solution for the original instance. Then, also the adapted instance with the inserted gadget does not provide an exact cover. To be more precise, look at the vertex v. Instead of S_v it could also be covered by S_1. But then, either S_2 or S_3 has to cover x_2 and thus, x_1 or x_3 would be covered by two subsets. Therefore, S_1 cannot cover v. Due to symmetry, S_3 cannot cover w as well. Other nodes than v and w are not affected by the new gadget. Altogether, we conclude

Theorem 2.5. *The* MAXIMUM LEAF SPANNING TREE *Problem is $\mathcal{NP}$-complete on graphs that are planar, cubic, and biconnected.*

Remark 2.6. *The preprocessing which was done to avoid element nodes of degree 1 can also be performed prior to the reduction of* P-X3C *to the* MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS *Problem, which is presented in Section 3.5. This problem is more interesting on biconnected graphs since the size of a strictly fundamental cycle basis is the sum of the sizes on the biconnected components.*

2.4 $\mathcal{APX}$-Completeness of the MLST on Regular Graphs

This section contains a proof for the $\mathcal{APX}$-completeness of the MLST on 5-regular graphs. For this, the MINIMUM DOMINATING SET Problem, restricted to cubic graphs, is L-reduced to the MLST Problem, restricted to 5-regular graphs. The approach of this reduction is based on [51]. The treated problems are defined in Subsection 2.4.1. In Subsection 2.4.2, we firstly present three necessary graph components and describe their properties. Afterwards, we construct functions f and g and derive constants α and β as introduced in Definition 1.14.

2.4.1 The Problems

We start with the denotation of the optimization version of the MLST Problem on 5-regular graphs. Remember that in Definition 1.14, we used the problem's name as index for the entries in the 4-tuple which constitutes the optimization problem. For these indices, we abbreviate 5-MLST with L.

5-MLST	
Instance:	5-regular graph $G_L = (V_L, E_L)$.
Feasible solution:	Spanning tree T of G_L.
Objective function:	Number of leaves of T, i.e. $\ell(G_L, T)$.
opt:	max.

The problem from which we reduce 5-MLST to prove it to be $\mathcal{APX}$-hard is the MINIMUM DOMINATING SET Problem restricted to cubic graphs. Therein, it is suitable to define $d(G_D, U)$ as the size of a dominating set U in a graph G_D, where the index D is short for 3-MDS.

3-MDS	
Instance:	Cubic graph $G_D = (V_D, E_D)$.
Feasible solution:	A set $U \subseteq V_D$, such that for all $v \in V_D \setminus U$ there is a node $u \in U$ with $uv \in E_D$.
Objective function:	$d(G_D, U)$.
opt:	min.

The 3-MDS is $\mathcal{APX}$-complete, see [4]. As there, we will use an L-reduction from 3-MDS to 5-MLST instead of an AP-reduction in order to prove that 5-MLST is $\mathcal{APX}$-hard. In this case, this is possible due to Lemma 1.15.

2.4.2 The Transformation

In this subsection, the transformation from 3-MDS to 5-MLST is presented, which is motivated by the transformation given in [51]. We start with the description of the gadgets, followed by functions f and g, constants α and β, as well as the necessary corresponding statements and their proofs.

The P-Components. Figure 2.10 shows the three required P-components. The maximum number of leaves at nodes with degree 5 is $8 - i$ for P^i, as one can see by taking a closer look at Figure 2.10. On the other hand, each component provides at least one leaf. From these components, we construct the instance for 5-MLST which is computed by the function f. Denote by $V(P^i)$ the set of nodes of component P^i, where the squared nodes are excluded, by $E(P^i)$ the set of edges of P^i which are incident to at most one squared node, and by $r(P^i)$ the grey node in P^i.

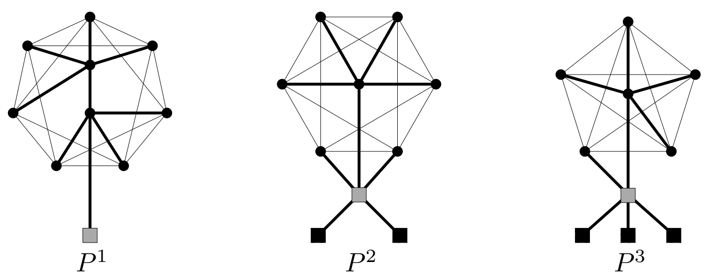


Figure 2.10: The three required components and highlighted spanning trees with the maximum number of leaves restricted to degree 5 nodes.

$f : \mathcal{I}_D \to \mathcal{I}_L$. The function f maps an instance of 3-Mds to an instance of 5-Mlst. For the rest of this section let $n = |V_D|$ and denote $V_D = \{v_1, \ldots, v_n\}$. Then define

$$\begin{aligned} V_L &= \bigcup_{i=1}^{n} \big(V(P_i^1) \cup \{x_i, y_i, z_i\}\big) \;\cup\; V(P_1^2) \;\cup\; \bigcup_{i=2}^{n-1} V(P_i^3) \;\cup\; V(P_n^2) \;, \\ E_L &= \bigcup_{i=1}^{n} \big(E(P_i^1) \cup \{x_i y_i, y_i z_i\}\big) \;\cup\; \bigcup_{i=1}^{n-1} \{z_i z_{i+1}\} \;\cup\; E(P_1^2) \;\cup\; \bigcup_{i=2}^{n-1} E(P_i^3) \;\cup\; E(P_n^2) \\ &\quad \cup \bigcup_{v_i v_j \in E_D} \{x_i y_j\} \;, \end{aligned} \tag{2.3}$$

and identify x_i with $r(P_i^1)$ for $i \in \{1, \ldots, n\}$, z_i with $r(P_i^2)$ for $i \in \{1, n\}$, as well as z_i with $r(P_i^3)$ for $i \in \{2, \ldots, n-1\}$. The set in Line (2.3) expresses that there is an edge between x_i and x_j if and only if there is one between v_i and v_j in the original graph. See Figure 2.11 for this setting, in which the P-components and the edges in $\bigcup_{v_i v_j \in E_D} \{x_i y_j\}$ are only indicated. The graph G_L is 5-regular, where the regularity at the x- and y-nodes is due to the 3-regularity of G_D. Further, this transformation can be done in polynomial time.

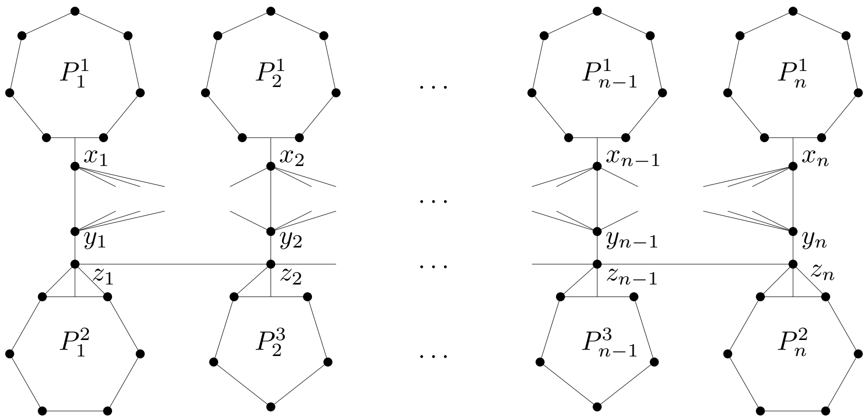


Figure 2.11: Indicated example of an instance computed by the function f.

$g : \mathcal{I}_D \times S_L \to S_D$. Now we describe the function g which computes a feasible solution for 3-Mds from a feasible solution of 5-Mlst on an instance transformed by f. For this, let T be an arbitrary spanning tree on such an instance. Remark that none of the x- or z-nodes can be leaves, since they are articulation points. Now, add the edges of $\bigcup_{i=1}^{n-1} \{z_i z_{i+1}\} \setminus T$ one after the other to T and call the resulting circuit C. For each C, delete the edge in $\bigcup_{v_i v_j \in E_D} \{x_i y_j\} \cap C$ with the minimum index j at the y-node. Observe that some of the y-nodes can become new leaves. Call the generated tree T'. Next, successively join the edges in $\bigcup_{i=1}^{n} y_i z_i \setminus T'$ to the tree, eliminate each time the unique edge $x_j y_i$ in the resulting circuit, and denote the constructed tree by T''. Note that T'' has the same set of leaves as

T', because the degrees of the y-nodes remain unchanged while all x- and all z-nodes are articulation points and hence cannot be leaves.

Define $g(G_D, T) = \{v_i \in V_D \mid \deg_{T''}(y_i) \geq 2\} =: N$, i.e. the set of the nodes in V_D corresponding to the y-nodes that are non-leaves in the tree T''. Again, this procedure can be done in polynomial time. Additionally, N is a dominating set in the original graph, since each P^1-component must be reachable from the P^2- and P^3-components via a y-node which is not a leaf. Thus, N is a feasible solution for the underlying 3-MDS instance.

The Constants. It remains to show that there are two constants α and β as demanded in Definition 1.14. For this, let $G_L = f(G_D)$.

Proposition 2.7. *With n being the number of nodes in G_D, it holds that $n \leq 4 \cdot \min(G_D)$.*

Proof. This is true since in a cubic graph, each vertex can dominate at most three other nodes. □

Lemma 2.8. *The maximum number of leaves of a spanning tree on G_L is $\max(G_L) = 13n + 2 - \min(G_D)$.*

Proof. We start to construct a spanning tree T of G_L with $13n + 2 - \min(G_D)$ leaves and show that a tree on G_L cannot contain more leaves. Observe that there are n components P^1, two components P^2 and $n - 2$ components P^3, which together contribute at most $n \cdot 7 + 2 \cdot 6 + (n - 2) \cdot 5 = 12n + 2$ leaves. Further leaves can only be among the y-nodes, because neither x- nor z-nodes can be leaves. To achieve a preferable large number of leaves, append the edges $\bigcup_{i=1}^{n-1}\{z_i z_{i+1}\} \cup \bigcup_{i=1}^{n}\{y_i z_i\}$ and $\bigcup_{j=1}^{n}\{x_j y_i \mid v_j \text{ is dominated by } v_i\}$ to the tree. Note that a vertex in a dominating set of G_D also dominates itself. The other $n - \min(G_D)$ y-nodes remain leaves, thus, T has $13n+2-\min(G_D)$ leaves, at all. Conversely, assume that there was a tree with more leaves. Since there cannot be more leaves in the P-components and the x- and z-nodes are articulation points, only the y-nodes could provide more leaves. But then, consider again the set $N = \{v_i \in V_D \mid \deg_{T''}(y_i) \geq 2\}$. If there were more leaves at the y-nodes, there was a dominating set N for G_D of size less than $\min(G_D)$, a contradiction. □

Lemma 2.9. *For all instances G_D it holds that $\max(G_L) \leq 53 \cdot \min(G_D)$.*

Proof. The proof is straightforward:

$$\begin{aligned}
\max(G_L) &\overset{\text{Lem. 2.8}}{=} 13n + 2 - \min(G_D) \\
&\overset{\text{Prop. 2.7}}{\leq} 13 \cdot 4 \min(G_D) + 2 - \min(G_D) \\
&\leq 53 \cdot \min(G_D).
\end{aligned}$$

□

Lemma 2.10. *For all spanning trees T of G_L it holds that* $|\min(G_D)-d(G_D,g(G_D,T))| \leq |\max(G_L)-\ell(G_L,T)|$.

Proof. Observe at first that

$$d(G_D,g(G_D,T)) = |N| \geq \min(G_D) \text{ and} \tag{2.4}$$
$$\max(G_L) \geq \ell(G_L,T). \tag{2.5}$$

Furthermore, it is

$$\ell(G_L,T) \leq 13n+2-|N|, \tag{2.6}$$

since at most $12n+2$ leaves are provided by the $2n$ P-components and there are $n-|N|$ leaves among the y-nodes. Thus, we derive

$$\begin{array}{rrcl}
 & |N|+\ell(G_L,T) & \leq & |N|+\ell(G_L,T) \\
\overset{(2.6)}{\Longrightarrow} & |N|+\ell(G_L,T) & \leq & |N|+13n+2-|N| \\
\Longleftrightarrow & |N|-\min(G_D) & \leq & 13n+2-\min(G_D)-\ell(G_L,T) \\
\overset{\text{Lem. 2.8}}{\Longleftrightarrow} & |N|-\min(G_D) & \leq & \max(G_L)-\ell(G_L,T) \\
\overset{(2.4)\text{ and }(2.5)}{\Longleftrightarrow} & |\min(G_D)-d(G_D,g(G_D,T))| & \leq & |\max(G_L)-\ell(G_L,T)|, \quad (2.7)
\end{array}$$

where Line (2.7) is exactly the fourth item in Definition 1.14 for $\beta=1$. □

After describing the polynomial time computable functions f and g and deriving $\alpha=53$ in Lemma 2.9 and $\beta=1$ in Lemma 2.10, we obtain an L-reduction from G_D to G_L. Hence, the 5-MLST is $\mathcal{APX}$-hard. Keeping in mind that there exists a 2-approximation for MLST on *arbitrary* graphs ([118]), we conclude

Theorem 2.11. *The* MAXIMUM LEAF SPANNING TREE *Problem is $\mathcal{APX}$-complete on 5-regular graphs.* □

2.4.3 Extension to Graphs with Arbitrary Odd Regularity

Now we show how the transformation in the last subsection can be extended to k-regular graphs for odd $k>5$. This proves the $\mathcal{APX}$-completeness for the k-MLST Problem, i.e. the MLST restricted to k-regular graphs with odd regularity. We use k instead of L as the index for the instance. Note that k itself does not belong to the instance. At next, we introduce a further component, the *hammock* H^k, see Figure 2.12 for $k=7$.

The general hammock H^k is constructed as follows. At first, create a path $(r(H^k), r_2(H^k), r_1(H^k))$. The node $r(H^k)$ is called the *root* of the hammock. Then, for $i\in\{1,2\}$, connect the nodes $r_i(H^k)$ to disjoint sets R_i of $k-i$ independent vertices. Now, link each node of R_1 to each one of R_2 such that the induced graph $H^k[R_1\cup R_2]$ is the bipartite graph

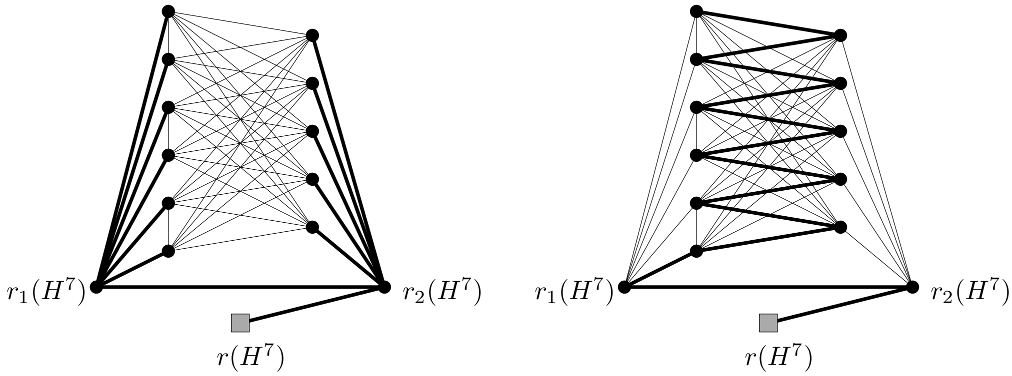


Figure 2.12: The hammock H^7 and highlighted spanning trees with $2k-3=11$ leaves resp. with 1 leaf.

$K_{k-1,k-2}$, at this point. The cardinality of R_1 is even, and thus, we can connect nodes pairwise in this set. After this step, each vertex besides $r(H^k)$ in the hammock has degree k. A spanning tree restricted to a hammock has at least 1 and at most $2k-3$ leaves, since $|V(H^k)| = 2k-2$ and $r(H_k)$ is an articulation point. We proceed with the description of the functions f and g and the constants α and β.

$\boldsymbol{f : \mathcal{I}_D \to \mathcal{I}_k}$**.** For $k > 5$, only this component is assembled into the instance computed by the function f, instead of the three different components in the case $k = 5$. Take again a look back at Figure 2.11 and consider the graph induced by the vertex set $\bigcup_{i=1}^{n}\{x_i, y_i, z_i\}$, i.e. the instance without the P-components.

For the node degrees in this truncated graph it holds that $\deg(x_i) = 4$, $\deg(y_i) = 5$ for $i \in \{1,\ldots,n\}$, $\deg(z_i) = 3$ for $i \in \{2,\ldots,n-1\}$, and $\deg(z_1) = \deg(z_n) = 2$. These degrees can be increased by appending a suitable number of hammocks. This cannot be done at the y-nodes, because some of them have to remain leaves. Therefore, the instance is modified in the following way.

For $i = 1,\ldots,n$ add $k-5$ vertices $u_i^1,\ldots,u_i^{k-5}$ and edges $x_iu_i^1, y_iu_i^1,\ldots,x_iu_i^{k-5}, y_iu_i^{k-5}$. Now, the y-nodes have degree k. At the x- and the z-nodes, the appropriated number of hammocks is attached by identifying these nodes with the hammock's root. In detail, glue one hammock at each x-node, $k-2$ hammocks at the u-nodes and at z_1 and z_n, as well as $k-3$ hammocks at $z_2,\ldots,z_{n-1}$. See Figure 2.13 which illustrates the described construction.

Observe that all nodes have degree k. Since all u-, x-, and z-nodes are articulation points, only y-nodes and nodes in hammocks can be leaves. As in the proof for $k=5$, exactly these y-nodes which correspond not to nodes of the dominating set can be chosen as leaves.

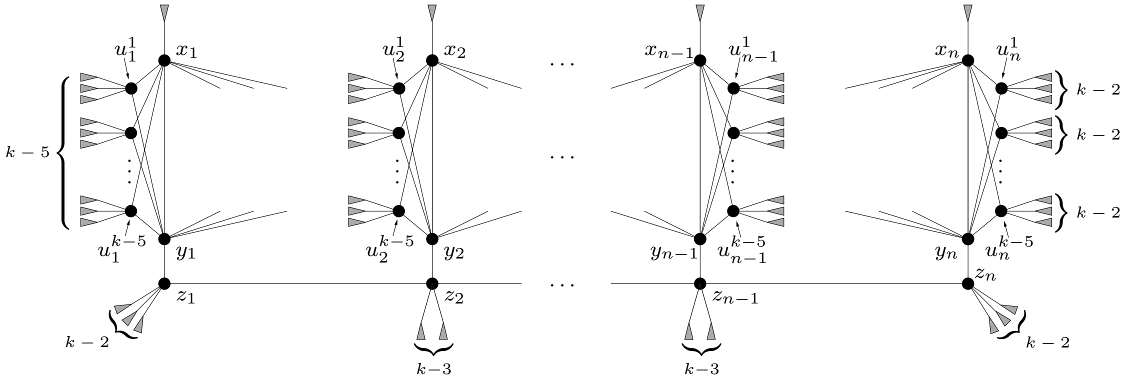


Figure 2.13: Construction of the k-regular instance for odd $k > 5$. The hammocks are depicted as grey triangles.

$\boldsymbol{g : \mathcal{I}_D \times S_k \to S_D}$. The description of this function can be directly carried over from the case $k = 5$.

The Constants. At a first glance, it seems to be confusing that the constants are dependable on k. But remember that k does not belong to the instance. Thus, after fixing k, they are constants as required. Now, we adapt Proposition 2.7 and Lemmas 2.8, 2.9, and 2.10. Again, let $G_k = f(G_D)$.

Proposition 2.12. *With $n = |V(G_D)|$, it holds $n \leq (k+1)\min(G_D)$.* ☐

Lemma 2.13. *The maximum number of leaves of a spanning tree on G_k is $\max(G_k) = (4k^4 - 36k^3 + 113k^2 - 150k + 73)n + 8k^2 - 24k + 18 - \min(G_D)$.*

Proof. The proof is essentially the same as the proof for Lemma 2.8. Anyway, we should break down the number of hammocks in G_k. There are n hammocks at the x-nodes, $(n-2)(k-3)$ at $z_2, \ldots, z_{n-1}$, $2(k-2)$ at z_1 and z_n, as well as $n(k-5)(k-2)$ hammocks at the u-nodes. All in all, there are $(2k^3 - 15k^2 + 34k - 24)n + 4k - 6$ hammocks. Since each hammock contributes at most $2k - 3$ leaves, the maximum number of leaves restricted to the hammocks is $(2k-3)((2k^3 - 15k^2 + 34k - 24)n + 4k - 6) = (4k^4 - 36k^3 + 113k^2 - 150k + 72)n + 8k^2 - 24k + 18$. Adding $n - \min(G_D)$, i.e. the maximum possible number of leaves restricted to the y-nodes, leads to the statement. ☐

Lemma 2.14. *For all instances G_D it holds $\max(G_k) \leq (4k^5 - 32k^4 + 77k^3 - 29k^2 - 101k + 90)\min(G_D)$.*

Proof. Use Lemma 2.13 and Proposition 2.7 as in the proof of Lemma 2.9. ☐

Lemma 2.10 and its proof can directly be transferred to deduce $\beta = 1$. Together with $\alpha = 4k^5 - 32k^4 + 77k^3 - 29k^2 - 101k + 90$ we get completely analogously to Theorem 2.11

Theorem 2.15. *The k-MLST is $\mathcal{APX}$-complete for odd $k \geq 5$.* ☐

2.5 Algorithms for Selected Graph Classes

In this section, we provide several restricted classes of graphs, on which the MAXIMUM LEAF SPANNING TREE Problem is solvable in polynomial time, or admits a PTAS, at least. Therein, we profit from the relationship of the MLST to the DEGREE PRESERVING SPANNING TREE Problem (DPST) on cubic graphs. We start with the definition of this problem, which we formulate as an optimization problem. This is followed by the description of its relationship to the MLST and to the definitions of several graph classes. The section is closed with a small table which summarizes the results.

DPST	
Instance:	Graph $G = (V, E)$.
Feasible solution:	Spanning tree T of G.
Objective function:	Number of vertices with *full degree* in T,
	i.e. $p(G,T) := \|\{v \in V \mid \deg_T(v) = \deg_G(v)\}\|$.
opt:	max.

A practical application of the DEGREE PRESERVING SPANNING TREE Problem is to measure flow in a water-distribution network. The flow in the network can be determined by examining only the chords of a spanning tree on the network. For such a chord, the pressure is measured at its incident vertices. Thus, one is interested in a spanning tree, whose chords are incident to a minimum number of vertices. On the other hand, such a tree has the maximum number of vertices with full degree. We refer to [21] for a more detailed description of this application and to [103] for the technical background.

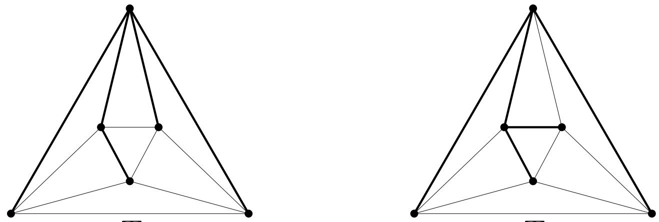
When we restrict ourselves to cubic graphs, we observe the following relationship.

Observation 2.16. *The* DPST *and the* MLST *are equivalent on cubic graphs.* □

To see this, denote $d_T^i = |\{v \in V \mid \deg_T(v) = i\}|$ for an arbitrary spanning tree T. Counting the vertices yields $n = d_T^1 + d_T^2 + d_T^3$, while counting the tree edges leads to $2(n-1) = d_T^1 + 2d_T^2 + 3d_T^3$. From these equalities we derive that $d_T^1 = d_T^3 + 2$ holds for cubic graphs. The statement on the equivalence of the problems restricted to cubic graphs is obtained by keeping in mind that T has exactly d_T^1 leaves and exactly d_T^3 nodes of full degree, and that their difference is always 2.

Note that this approach is not transferable to k-regular graphs for $k > 3$. For an illustration take a look below at Example 2.17.

Example 2.17.
Consider the 4-regular octahedron graph G_O in Figure 2.14 and the spanning trees T_1 and T_2. On the one hand, we have $d_{T_1}^1 = d_{T_2}^1 = 4$, and on the other hand, $p(G_O, T_1) = 1 \neq 0 = p(G_O, T_2)$. This shows that the maximum number of leaves of a tree does not depend by a constant on the number of vertices with full degree in the tree. ♢



T_1 T_2

Figure 2.14: Octahedron graph G_O with highlighted spanning trees T_1 and T_2.

In [21], the authors provide several results concerning the Degree Preserving Spanning Tree Problem for some classes of graphs. Due to Observation 2.16, these results can directly be transferred to the Mlst if the particular graph class is additionally required to be cubic. We shortly define the discussed graph parameters and classes of graphs and summarize the appropriated results below.

Asteroidal Number (AN). A set $\{v_0, v_1, v_2\} \subset V$ of three independent vertices in a graph $G = (V, E)$ forms an *asteroidal triple* if there is a v_i-v_{i+1}-path in $G[V \setminus N[v_{i+2}]]$, where the indices are modulo 3. A set $A \subseteq V$ is an *asteroidal set* if each triple of vertices in A is asteroidal. The *asteroidal number* $\mathrm{an}(G)$ is the maximum cardinality of an asteroidal set in the graph G. Asteroidal sets as a generalization of asteroidal triples were introduced in [121] for the characterization of subclasses of *chordal graphs*, i.e. graphs without induced C_k for each $k \geq 4$.

Treewidth. We decided to omit the usual definition of treewidth in terms of a tree-decomposition. For our purpose, it is sufficient to state that a graph G has *treewidth* $\mathrm{tw}(G) \leq k$ if and only if it is a subgraph of a chordal graph with clique number $\omega(G) \leq k + 1$. For the relationship with tree-decomposition see e.g. [16].

Interval Graphs. A graph is an *interval graph* if it is isomorphic to a graph $G = (V, E)$ with $V \subset \{[a, b] \subseteq \mathbb{R} \mid a \leq b\}$ and $E = \{\{u, v\} \in V^2 \mid u \cap v \neq \emptyset\}$.

Cocomparability Graphs. An undirected graph G is a *comparability graph* if there is an orientation $D = (V, A)$ of G with $\{(x, y), (y, z)\} \subseteq A \Rightarrow (x, z) \in A$. A graph is a *cocomparability graph* if it is the complement of a comparability graph.

2.6 Conclusions and Outlooks

This chapter was involved with the Maximum Leaf Spanning Tree Problem. We were able to show the $\mathcal{NP}$-completeness of this problem on cubic planar graphs which are additionally biconnected—all in all a very specific graph class. An obvious enhancement of this work would be to show the $\mathcal{NP}$-completeness of this problem for planar graphs that are regular of degree 4 or 5. Let us shortly consider 4-regular graphs, at first. Having in

graph class	running time	remarks
cubic graphs with bounded AN	$\mathcal{O}(2^{k^3} n^{k+3} \log n)$	$k = \mathrm{an}(G)$
cubic graphs with bounded treewidth	$\mathcal{O}(n)$	preceding computation of $\mathrm{tw}(G)$
cubic interval graphs	$\mathcal{O}(n)$	
cubic cocomparability graphs	$\mathcal{O}(n^4)$	
cubic planar graphs	-	PTAS ([14])

Table 2.1: Summary of the derived running times of the MLST for several selected graph classes.

mind that the switcher (Figure 2.6) has 5 outgoing edges, we observe that we won't get a 4-regular version of this subgraph. Also the gear might be designed slightly different. Nevertheless, we conjecture that the problem is also $\mathcal{NP}$-complete on 4- and 5-regular planar graphs, but the design of the components could be more sophisticated. We note that our result can be helpful to prove the $\mathcal{NP}$-completeness of other interesting and applicable problems, especially on planar or cubic graphs.

A further result was the $\mathcal{APX}$-completeness of the MLST on 5-regular graphs. The proof could be extended to k-regular graphs for all odd $k > 6$. This result is only slightly different to what the authors of [20] and [28] had desired: the max $\mathcal{SNP}$-completeness of the MLST for cubic graphs. Additionally, the search for approximation algorithms for MLST on k-regular graphs with $k > 3$ would be interesting.

After observing the equivalence of the MAXIMUM LEAF SPANNING TREE Problem and the DEGREE PRESERVING SPANNING TREE Problem on cubic graphs, several polynomial time algorithms for the latter problem can also be used for the first one on 3-regular graphs. In particular, the maximum leaf number can be computed in polynomial time on cubic graphs with bounded asteroidal number or with bounded treewidth, on cubic interval graphs and on cubic cocomparability graphs. A PTAS for cubic planar graphs—that class for which we initially proved the $\mathcal{NP}$-completeness of the MLST—exists as well. Here, further lines of research could be the design of polynomial time algorithms or PTASs for more classes of graphs, possibly less restricted ones.

Chapter 3

Strictly Fundamental Cycle Bases

In this chapter, we investigate strictly fundamental cycle bases, SFCB for short. As introduction to the topic, we provide short outlines of the history, complexity and approximability considerations, and of a priori upper bounds. In Section 3.2, we describe three applications for which the usage of strictly fundamental cycle bases is necessary or advantageous. The basic definitions and some elementary properties of strictly fundamental cycle bases in general are illustrated in Section 3.3. This is specified to strictly fundamental cycle bases on planar graphs in Section 3.4, where we also define planar graphs in a stringent way. Moreover, some important properties of strictly fundamental cycle bases are generalized to weighted planar graphs in order to describe the relationship of the MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS Problem (MSFCB) with the OPTIMUM COMMUNICATION SPANNING TREE Problem. Section 3.5 is dedicated to the complexity of the MSFCB on planar graphs. The main part of Section 3.6 on SFCBs on outerplanar graphs is the investigation of the minor monotonicity of minimum SFCB weight on weighted outerplanar graphs. In Section 3.7, we present cycle root graphs, i.e. a class of weighted graphs on which the MSFCB is solvable in polynomial time.

Contribution. Deo et al. ([32]) established the $\mathcal{NP}$-completeness of the MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS Problem for general graphs. A similar proof for the $\mathcal{NP}$-completeness on bipartite graphs was given in [78]. Beside these results, no other class is known on which the MSFCB is $\mathcal{NP}$-complete. Especially for planar graphs, the question about the complexity status of the MSFCB restricted to this class was pointed out in several publications, e.g. in [64, 78, 129]. In Section 3.5, we provide an $\mathcal{NP}$-completeness proof by reducing the planar version of the EXACT COVER BY 3-SETS Problem. The reduction is similar to the one given in Section 2.3. Restricted to planar graphs, the MSFCB was the only minimum cycle basis problem whose complexity status was unknown. All other problems are solvable in polynomial time, where minimizing among weakly fundamental, totally unimodular, and integral cycle bases coincide.

Minor monotonicity plays an important role when deriving an a priori upper bound for the size of a minimum strictly fundamental cycle basis on an unweighted outerplanar graph, see [108]. This behaviour is already lost when we consider weighted outerplanar graphs. In Section 3.6, we define a subclass of weighted outerplanar graphs on which the minimum strictly fundamental cycle basis weight is actually minor monotone. Simultaneously, the minimum weight of a strictly fundamental cycle basis is computable in polynomial time on this class. A second class of weighted graphs with this property is provided in Section 3.7.

3.1 Introduction

In Chapter 2, we investigated the problem of finding a spanning tree with as many leaves as possible. Another spanning tree problem is to search for a small strictly fundamental cycle basis of a graph. For a given graph and a spanning tree on it, the insertion of one further edge of the graph to this tree produces a unique circuit, a *fundamental circuit*. The set of all fundamental circuits with respect to a spanning tree of a graph forms a *strictly fundamental cycle basis*. For a formal definition and further characterizations we refer to Section 3.3.

Historically, strictly fundamental cycle bases go back to a definition of Kirchhoff in 1847 ([65]), which is why SFCBs are sometimes called *Kirchhoff bases* or *Kirchhoff-fundamental bases* by several authors, e.g. [13,67]. The term *fundamental* itself seems to be used for the first time in [126], even though in the more general context of matroids. Because of their early appearance, the straightforward way to construct them, and their simple structure, the class of strictly fundamental cycle bases is maybe the most prominent one among the classes of cycle bases studied in this thesis. On the other hand, when leaving aside the robust cycle bases, which are the topic of Chapter 4, the class of SFCBs is also the most specialized one.

The Minimum Strictly Fundamental Cycle Basis Problem seeks for a spanning tree whose fundamental circuits constitute an SFCB that is as short as possible. Despite the simple nature of SFCBs, this problem turns out to be hard, similarly as many other easily appearing combinatorial problems. In its unweighted version, the problem is $\mathcal{NP}$-complete on general graphs ([32]) and on bipartite graphs, cf. [78]. Both proofs, which are reductions from the Shortest Total Path Length Spanning Tree Problem, have a profoundly non-planar structure, and it does not seem to be possible to convert them to a proof for planar graphs. In Section 3.5, we proof that the Msfcb is also $\mathcal{NP}$-complete on planar unweighted graphs by providing a reduction from another $\mathcal{NP}$-complete problem. This result is contrary to the legitimate statement in [78] that "theoretical, the Minimum Strictly Fundamental Cycle Basis Problem still could turn out to be polynomially solvable on planar graphs ...". The question of the complexity status of the Msfcb on planar graphs appeared also as Problem 14 in the Survey [64].

The complexity status for weighted outerplanar graphs is unknown. The problem is solvable in linear time on unweighted outerplanar graphs by performing a breath first search on the dual graph. Galbiati et al. ([52]) investigated the approximability of the MSFCB. The problem is $\mathcal{APX}$-hard for general unweighted graphs. It is approximable within $\mathcal{O}(\log^2 n \log\log n)$ for arbitrary weighted graphs and within $\mathcal{O}(1)$ ($\mathcal{O}(\log n)$) for weighted graphs with $\binom{n}{2} - m \in \mathcal{O}(1)$ ($\binom{n}{2} - m \in \mathcal{O}(\log n)$). Moreover, there is a polynomial-time approximation scheme for the MSFCB on complete weighted graphs.

Upper bounds for the minimum length of an SFCB can be derived from [1]. More precisely, the minimum SFCB of a graph with a metric weight function has a size of $\mathcal{O}(m \log n \log^3 \log n)$. This bound decreases to $\mathcal{O}(m \log n \log\log n \log^3 \log\log n)$ if the graph is unweighted. However, these bounds might appear rather unnatural or artificial. Since the size of a minimum SFCB of an unweighted planar grid graph is in $\Theta(n \log n)$—see [68] for lower bounds and [82] for the best known upper bound—we conjecture that also for arbitrary graphs a minimum SFCB has a size of $\mathcal{O}(m \log n)$.

3.2 Applications

As mentioned above, strictly fundamental cycle bases form the most specialized class of cycle bases. Thus, if any application asks for a cycle basis of a defined type, an SFCB will also be suitable. Note that this strategy may not work if a class of robust cycle bases is demanded, or if one is interested in a cycle basis of a certain type which is actually minimum. Clearly, a cycle basis which is not strictly fundamental could be smaller than a minimum SFCB.

In this section, we pick out three applications for which strictly fundamental cycle bases are indispensable or have essential advantages over other classes.

Graph Drawing. When drawing a graph, one is usually interested in a clear and esthetical representation. The term *esthetical* is surely subjective but there are several criteria for drawing a graph in an esthetical manner. An obvious criterion is the minimization of the number of edge crossings. Moreover, edges should be straight and rather short lines, and the angle between two edges at one vertex ought not to be too small (see [45] for an overview of such criteria). From now on, we focus on *clustered* representations in this paragraph.

As one would guess, in a real-world graph—like, e.g., a social network—the edges of this graph are not uniformly distributed. In the special case of a social network, parts of the graph tend to establish nearly complete subgraphs, which we refer to as *clusters*. Note that we do not need a stringent definition of clusters at this place, but we remark that the diameter of the subgraph induced by a cluster should be rather small. Anyway, it turned out that strictly fundamental cycle bases are useful for an acceptable depiction of a clustered graph, see e.g. [72]. There, these observations are also encouraged by experimental results.

We give only a rough idea of how to use SFCBs for the construction of an esthetical layout of a graph. Consider the inducing tree of a small strictly fundamental cycle basis and a certain cluster. Then, this tree restricted to the cluster is often a subtree. Otherwise, many non-tree edges in this cluster induce large circuits since the paths between their end vertices have to leave and to re-enter the cluster. Thus, one computes a spanning tree that induces a small strictly fundamental cycle basis and weights the edges with the tree distance of their end vertices to each other. Then, one uses this updated graph as the input for a spring embedding algorithm. Such an algorithm embeds a graph as if it was a physical system of springs. Note that although some parts of the graph might be locally planar, a plane embedding of these parts cannot be expected in general. We refer to [49] for a more detailed description of spring embedding algorithms.

Electrical Engineering. In the context of electrical engineering, Kirchhoff's voltage law states that in an electrical network the potential differences along any circuit add up to zero. More formally, if p is the vector which contains the potential differences of all arcs in the directed network, then $\langle C, p\rangle = 0$ holds for each circuit C of the network's cycle space $\mathcal{C}$. If B is a strictly fundamental cycle basis of $\mathcal{C}$, it is merely sufficient that the equation $\langle C, p\rangle = 0$ holds for all circuits in the cycle basis B.

Of course, the last sentence is also valid if B is a cycle basis which is not necessarily strictly fundamental. But for the special application in electrical engineering, SFCBs should be preferred due to a higher numerical stability of following calculations ([68]). Much more elaborate descriptions of how to express Kirchhoff's voltage law in terms of cycle bases can be found in [17] and in [18]. For the numeric analysis see for example [9].

Chemistry. The application of graph theory to chemistry is very old. It dates back to 1875 at least, when Cayley studied trees and used them to enumerate isomers of chemical compounds, see [22]. As is known, the famous Cayley's formula for the number of spanning trees in a labeled complete graph is due to him, as well.

Amongst others, chemical applications of graph theory are to model molecules and networks of chemical reactions. We only want to describe the former application. For chemists, the cyclic structure of a molecule and hence of its modeling graph often is interesting. In practise, it is used for example for the retrieval of molecules in chemical databases. To minimize the retrieval time, *smallest sets of smallest rings (SSSR)* crystallized to be useful. Unfortunately, SSSRs are not consistently defined in chemical literature. So it is defined equivalently to minimum cycle bases, to minimum weakly fundamental cycle bases, and to minimum strictly fundamental cycle bases. Clearly, this inconsistency can lead to confusion. Anyway, the use of SFCBs seems to be very helpful in this context. For further information on chemical applications of cycle bases we refer to [54] (especially Section 1.2), to [13], and, in both cases, to the references cited therein.

3.3 Basic Definitions and Properties on SFCBs

In recent years, various terms came up for what we call *strictly fundamental cycle bases*, and these terms did not always differ from the notion of *weakly fundamental cycle bases*. Thus, in this section, we provide a short survey of definitions and characterizations for *strictly* respectively *weakly* fundamental cycle bases, followed by an overview of several terms used for both classes of cycle bases. The section closes with three technical lemmas related to SFCBs on directed and on weighted graphs.

Let us start with the definition from Whitney which additionally contains the definition of weakly fundamental cycle bases.

Definition 3.1 (weakly/strictly fundamental cycle basis, [126]). *A set $B = \{C_1, \ldots, C_\nu\}$ of cycles in a directed graph D is a* weakly fundamental cycle basis *of D if there exists a permutation $\pi \in S_\nu$ such that*

$$C_{\pi(i)} \setminus \bigcup_{j=1}^{i-1} C_{\pi(j)} \neq \emptyset \quad \text{for all } i = 2, \ldots, \nu. \tag{3.1}$$

The set B is a strictly fundamental cycle basis *if (3.1) holds for every permutation $\pi \in S_\nu$.*

Definition 3.1 reflects a simple but useful property of weakly fundamental cycle bases. For one sequential arrangement of the basic circuits, it is possible to decompose the graph by gathering basic circuits from the graph, where gathering is only allowed by pulling the circuit at an arc which is not contained in another basic circuit. A decomposition in this way is possible if the cycle basis is weakly fundamental, since there exists an appropriate order to do this. This order does not play any role in the case of strictly fundamental cycle bases.

If each arc in a biconnected graph is contained in at least two basic circuits then this basis cannot be weakly fundamental since for each permutation, Inequality (3.1) does not hold. On the other hand, there are non-fundamental cycle bases of graphs, for which there are arcs that are member of only one basic circuit. Thus, the decision problem whether a given cycle basis is not weakly fundamental cannot be solved in $\Theta(m)$ time in a straightforward manner by simply checking if each arc is contained in two or more basic circuits.

Rather, we suggest the following routine to recognize a cycle basis as not weakly fundamental. Pick out an arc that is contained in only one basic circuit. Remove the arc from the graph and the circuit from the basis. Repeat this step. If this procedure ends in the empty graph, then the basis had been weakly fundamental. If one comes across a graph with a cycle basis where each arc is contained in more than one basic circuit, the basis had been non-fundamental. Hence, to decide whether a given basis is weakly fundamental by using the described algorithm depends linearly on the size of this basis.

If a cycle basis is not weakly fundamental, we can make another statement on the number of basic circuits in which an arc is contained.

Lemma 3.2 (cf. Lemma 10.9 in [78]). *If an undirected cycle basis B of a biconnected digraph D is not weakly fundamental, then there is an arc which is contained in at least three cycles of B.*

Proof. Initially, perform the routine described above, i.e. remove arcs that appear in only one basic cycle C and remove C from B. It remains a graph $D' = (V', A')$ with a cycle basis $B' = \{C_1, \ldots, C_{\nu'}\}$ for which each arc is contained in at least two cycles of B'. Assume now that each arc is in exactly two cycles of B'. Then the projection $\pi(B')$ is not linear independent since $\sum_{C \in B'} \pi(C) = 0 \in \mathrm{GF}(2)^{|A'|}$. Hence, $\pi(B')$ is not a cycle basis of the underlying graph $G(D')$ and thus $\pi(B)$ is not a cycle basis of $G(D)$, a contraction to the condition that B is an undirected cycle basis. $\square$

Note that here it is really necessary to require a cycle basis which is undirected. In Section 1.2, we already mentioned Example 5 in the Survey. This example deals with a graph and a cycle basis in which each arc is actually contained in exactly two basic circuits. Clearly, this basis is not undirected.

The definition below constitutes a restriction to directed graphs of Kirchhoff's original definition for circuits in a matroid.

Definition 3.3 (Kirchhoff-fundamental cycle basis, [65]). *A cycle basis B of a directed graph $D = (V, A)$ is called* Kirchhoff-fundamental *if there exists a spanning tree $T = (V, A(T))$ of D with $B = \{C_T(a) \mid a \in A \setminus A(T)\}$. The term $C_T(a)$ denotes the unique circuit in $T + a$, the* fundamental circuit. *The tree T* induces *the basis B.*

Similarly to fundamental cycles, also *fundamental cuts* can be easily defined via a spanning tree. The set of all fundamental cuts with respect to some spanning tree analogously forms a basis of the cut space, see Section 1.2. To be accurate, let T be a spanning tree of a connected graph G and $e \in E(T)$. Then $T \setminus e$ consists of two connected components with node sets V_1 and V_2. The fundamental cut $S_T(e)$ is the set of all edges in G, including e, with one end node in V_1 and the other in V_2.

The definition of Kirchhoff-fundamental cycle bases shows the simple and straightforward possibility to construct, to display, and to store the cycle basis by using a spanning tree. Obviously, this spanning tree is not unique for a given SFCB in general (simply take graphs which contain only one circuit). Observe further, that for strictly fundamental and for Kirchhoff-fundamental cycle bases there is at least one edge per basic circuit C which is not contained in any other basic circuit. It is reasonable to think about exactly one edge with this property. Denote this very edge as *private edge* of the circuit C. In the case of Kirchhoff-fundamental bases the chords of the spanning tree should be viewed as private edges.

The definitions of weakly, strictly, and Kirchhoff-fundamental cycle bases can directly be applied to undirected graphs. Concerning the cycle matrix Γ which corresponds to a cycle basis of an undirected graph, strictly fundamental cycle bases can be characterized in a straightforward manner.

Lemma 3.4 (see e.g. [64]). *A cycle basis of an undirected graph is strictly fundamental if and only if the rows of Γ can be permuted such that Γ contains the $\nu \times \nu$ unit matrix in its first ν rows. In the case of directed graphs, some rows might be multiplied by -1 to yield this result.* □

The definitions of strictly fundamental and Kirchhoff-fundamental cycle bases turn out to be equivalent, see e.g. [78]. In the following, we prefer to use only the term *strictly fundamental cycle bases.* As already mentioned, the definition for strictly and weakly fundamental cycle bases is not unique in the literature.

The *fundamental cycle bases* of Hartvigsen and Zemel in [57] are equal to weakly fundamental cycle bases. SFCBs appeared there as a special case of their fundamental cycle bases. Using our terminology, the authors studied graphs for which every cycle basis is weakly fundamental. They were able to give three characterizations for such graphs. In [120], Sysło defined *sets of fundamental cycles* which are nothing else then SFCBs. He investigated cut graphs of cycle bases. Amaldi et al. ([6]) developed an "edge-swap" heuristic to locally minimize the size of a given SFCB. They simply denote SFCBs as *fundamental cycle bases.*

Now let us first turn our focus to directed graphs, and then to weighted graphs. In the case of directed graphs, a strictly fundamental cycle basis has a further property which involves the representation of a simple cycle as a linear combination of basic circuits. Actually, this looks similar to the characterizations for totally unimodular cycle bases, cf. Subsection 5.2.2.

Lemma 3.5. *Let D be a directed graph and B a strictly fundamental cycle basis of D. Then every simple cycle in D can be written as a linear combination of circuits in B with coefficients in $\{-1, 0, +1\}$.*

Proof. Let $S = \sum_{i=1}^{\nu} \lambda_i C_i$ be a simple cycle. Since the private arc of a basic circuit C_i appears λ_i times in S, it follows that $|\lambda_i|$ must be in $\{0, 1\}$. □

We conjecture that also the converse of Lemma 3.5 is true. If so, it would be a further nice characterization for SFCBs on directed graphs. In Chapter 4 about robust cycle bases we will need the following technical property of strictly fundamental cycle bases on weighted graphs.

Lemma 3.6 (Weight Lemma). *For a given strictly fundamental cycle basis B of an undirected graph $G = (V, E)$ one can always find a weight function w such that B is the unique minimum cycle basis of G.*

Proof. Let T be a fundamental spanning tree which induces B. For every edge $e \in T$ set $w(e) = 1$. Define $d := \max\{\mathrm{dist}_T(u, v) \mid uv \in E \setminus T\}$ and assign $w(e) = 2d - \mathrm{dist}_T(u, v)$ for the remaining edges $e = uv$. Observe that the minimum of w restricted to the chords is d. Now every circuit in B has a weight of $2d$ while all other cycles of G have a greater weight since they contain at least two chords and at least one tree edge or at least three chords. □

However, for Example 4.13, the usage of the Weight Lemma will not be sufficient. More detailed, $w(e) \geq \mathrm{dist}_T(u, v)$ does hold for each edge $e = uv$ in $E \setminus T$, when w is constructed according to the Weight Lemma. To obtain a *unique* minimum cycles basis after a modification of a given minimum basis in that example, it will be necessary that $w(e) < \mathrm{dist}_T(u, v)$ does additionally hold for one chord $e = uv$ of the spanning tree T. This is the statement of the next lemma.

Lemma 3.7 (Modified Weight Lemma). *For a given strictly fundamental cycle basis B of an undirected graph $G = (V, E)$ one can always find a weight function w such that B is the unique minimum cycle basis of G and such that there is a chord $e = uv$ with $w(e) < \mathrm{dist}_T(u, v)$.*

Proof. The proof has essentially the same structure as the proof of Lemma 3.6. Thus, set $w(e) = 1$ for all tree edges of a given fundamental spanning tree T which induces B. And again, let $d := \max\{\mathrm{dist}_T(u, v) \mid uv \in E \setminus T\}$. For the edges $e = uv$ in $E \setminus T$, we now assign the weight $w(e) = 2d - \mathrm{dist}_T(u, v) - \varepsilon$, for an $\varepsilon > 0$ whose value is determined later. The minimum of w restricted to the chords is $d - \varepsilon$, and each circuit $C \in B$ has the weight $w(C) = 2d - \varepsilon$.

Now, look at a circuit which is not in B. It consists of $c \geq 2$ chords and $t \geq 0$ tree edges. Furthermore, $c = 2$ implies $t \geq 1$. The length of the circuit is at least $c(d - \varepsilon) + t$. For all $\varepsilon \in (0, \frac{(c-2)d+t}{c-1})$, this value is greater than $2d - \varepsilon$, i.e. greater than the weight of a basic circuit. Because $c \geq 2$, the denominator of the upper endpoint of the interval is not zero, and since $c + t \geq 3$, also the numerator is not zero. Hence, this interval is not empty and we can take any ε from this interval.

Finally, for a chord $e = uv$ with $\mathrm{dist}_T(u, v) = d$, the weight $w(e)$ has the value $d - \varepsilon < \mathrm{dist}_T(u, v)$. □

3.4 SFCBs on Planar Graphs

Planar graphs, i.e. graphs that can be drawn without edge crossings into the plane, form one of the most prominent classes of graphs. At a first glance, a lot of real-world networks are planar or almost planar. When modeling diverse problems on these networks, they can lose their planarity in many cases. Nevertheless, it is not necessary to say that it is worth to investigate this class. Due to the high degree of popularity of planar graphs we skip further motivations and refer to standard literature on this topic—nearly every graph theory book contains a chapter on planar graphs. Instead, we pass on to a stringent definition of planar graphs in which we follow [70].

3.4.1 Definitions for Planar Graphs

In this subsection, we provide the formal definitions in the context of planar graphs. Therefore, we choose a topological approach. The reader is assumed to be familiar with the foundations of topology. The definitions of terms like the *interior* $\mathrm{int}(X)$ and the *boundary* $\partial(X)$ of a set X or *homeomorphism* can be found in standard topology books such as [104, 116]. All definitions can easily be applied to directed graphs.

Definition 3.8 (plane graph). *A graph $G = (V, E)$ is a* plane graph *if*

1. $V \subset \mathbb{R}^2$, $E \subset \mathcal{P}(\mathbb{R}^2)$, *where $\mathcal{P}(\mathbb{R}^2)$ denotes the set of all subsets of $\mathbb{R}^2$,*
2. *for each edge $e \in E$ there is a homeomorphism $\varphi_e : [0, 1] \to e$ with $\varphi_e(0), \varphi_e(1) \in V$,*
3. *for all pairs $e_1, e_2 \in E$ of edges it holds $\{\varphi_{e_1}(x) \mid x \in (0, 1)\} \cap (V \cup e_2) = \emptyset$.*

The first two items specify how the graph is drawn into the Euclidean plane, while the third one ensures that this drawing does not have edge crossings. For technical reasons, we extend the homeomorphism in item 2 to intervals. Thus, $\varphi_e(\mathcal{I}) := \{\varphi_e(x) \mid x \in \mathcal{I}\}$ for an interval $\mathcal{I} \subseteq [0, 1]$. For a given biconnected plane graph $G = (V, E)$, the set $\mathbb{R}^2 \setminus \bigcup E$ consists of (open) domains which are referred to as *faces* of G. Two faces f_1 and f_2 are *adjacent* if $|\partial(f_1) \cap \partial(f_2)| = \infty$, i.e. if their boundaries share at least one edge. There is exactly one unbounded face which is denoted by f^∞ and named the *exterior face*. All other faces are *interior faces*. An interior face which is not adjacent to f^∞ is an *internal face*. A plane graph without any internal face is called *internal face free*.

Definition 3.9 (planar graph, embedding, plane circuit, uniquely embeddable). *A graph $G = (V, E)$ is a* planar graph *if there exists a plane graph $G' = (V', E')$ which is isomorphic to G in virtue of a bijection $\varphi : V \to V'$. This plane graph is called an* embedding *of G. A* plane circuit *is a circuit C in G for which there is an embedding G' of G such that $\bigcup\{\varphi(u)\varphi(v) \mid uv \in C\}$ is the boundary of some face of this embedding. A biconnected planar graph G is* uniquely embeddable *if for all pairs $G_1 = (V_1, E_1)$*

and $G_2 = (V_2, E_2)$ *of isomorphic plane graphs with face sets* F_1 *and* F_2 *there exist two homeomorphisms* $\varphi^V : V_1 \to V_2$ *and* $\varphi^F : F_1 \to F_2$ *with* $v \in \partial(f) \Rightarrow \varphi^V(v) \in \partial(\varphi^F(f))$.

Using our terminology, a classical result of Whitney reads as follows:

Theorem 3.10 ([125]). *A 3-connected planar graph is uniquely embeddable.* □

Definition 3.11 (dual graph). *A plane graph* $G^* = (V^*, E^*)$ *with face set* F^* *is the* dual graph *of a given biconnected and plane graph* $G = (V, E)$ *with the set of faces* F*, if there exist bijections* $\varphi_V : V \to F^*$, $\varphi_E : E \to E^*$ *and* $\varphi_F : F \to V^*$ *with*

1. $v \in \varphi_V(v)$ *for all* $v \in V$,
2. $\varphi_F(f) \in f$ *for all* $f \in F$,
3. $|\varphi_e((0,1)) \cap \varphi_E(\varphi_e((0,1)))| = |e \cap \bigcup E^*| = |\varphi_E(e) \cap \bigcup E| = 1$ *for all* $e \in E$.

In this context, G *is called the* primal graph*, and it contains* primal vertices*,* primal edges*, and* primal faces*.*

Here, the first two items guarantee that each primal vertex lies in exactly one dual face and vice versa. The third item makes sure that each primal edge crosses exactly one dual edge and vice versa, and, on the other hand, it assures that edge crossings are only allowed between corresponding primal and dual edges.

Definitions 3.8 and 3.11 may appear somewhat unhandy, thus, we should dispense with them and restrict ourselves to a more convenient and intuitive interpretation. For the rest, it is sufficient just to imagine a graph to be drawn into the plane, although such a drawing is not unique in general. Nevertheless, all considerations take place on abstract planar graphs.

The dual graph is uniquely embeddable if the primal graph is. This is fulfilled if the primal graph is 3-connected (see, e.g. [36]). Note that this condition is not necessary since for example trees are also uniquely embeddable.

Definition 3.12 (dual tree). *For a given spanning tree* T *of a planar graph* $G = (V, E)$*, define the* dual tree T^* *of* T *in the dual graph* $G^* = (V^*, E^*)$ *of* G *as* $T^* = \{e^* \in E^* \mid \varphi_E^{-1}(e^*) \in E \setminus T\}$.

When we consider planar graphs in connection with cycle bases, we should mention the notion of *2-bases*. We will need them in Section 3.6 on outerplanar graphs.

Definition 3.13 (2-basis). *A cycle basis* B *of an undirected graph* $G = (V, E)$ *is called a* 2-basis *if each edge of* E *is contained in at most two cycles of* B.

Another denotation for a 2-basis is *simple basis*, which is used e.g. in [36]. Already in 1937, MacLane presented the following algebraic characterization for planar graphs by using 2-bases.

Theorem 3.14 ([87]). *A 2-connected undirected graph has a 2-basis B if and only if it is planar. In this case, each element in B is a plane circuit.* □

This gives rise to the term *planar basis* for a 2-basis in [54] what directly reflects the result of Theorem 3.14. In the context of fundamental bases, 2-bases relate as follows.

Lemma 3.15 (Lemma 10.12 in [78]). *Every 2-basis is weakly fundamental.* □

Note that it would not be difficult to directly carry over Definition 3.13 to digraphs. Actually, there is an example of a directed graph with a cycle basis B where each arc is contained in exactly two circuits of B. On the other hand, not each element in B is a plane circuit, see again Figure 1.1. Instead, a basis $B = \{C_1, \dots, C_\nu\}$ in a digraph D is a 2-basis if the condition holds that the set $\{\pi(C_1), \dots, \pi(C_\nu)\}$ is a cycle basis in $G(D)$, i.e. B is an undirected cycle basis.

Concerning the minimization among 2-bases of a planar graph G, one has to find a plane circuit C of maximum weight. Then embed the graph such that C is the boundary of the unbounded face. Then, the boundaries of the finite faces of this embedding form a minimum 2-basis. Finding a plane circuit of maximum weight is easy when the graph is 3-connected since in this case it is uniquely embeddable. If the planar graph is not triconnected and thus not uniquely embeddable in general, it is more difficult to find such a plane circuit. Nevertheless, it is tractable by the use of the $SPQR$-tree data structure, which had been introduced in [33]. Theorem 3.16 summarizes these results.

Theorem 3.16 (Theorem 33 in [81]). *The problem of finding a minimum 2-basis of a planar graph is solvable in $\mathcal{O}(n)$, possibly with weights on the edges.* □

Similarly, one may ask how minimum cycle bases restricted to other classes of cycle bases look like. Intuitively, they should consist only of circuits. The theorem below summarizes the known results.

Theorem 3.17 (Theorem 3.14 in [64]). *A minimum 2-basis consists only of circuits. This is also true for directed, undirected, weakly and strictly fundamental cycle bases.* □

This statement is trivial in the case of strictly fundamental cycle bases since there is no SFCB that contains a cycle which is not a circuit.

3.4.2 SFCBs on Weighted Planar Graphs

In this subsection, we study properties of strictly fundamental cycle bases on *weighted* planar graphs with a weight function $w : E \to \mathbb{R}_{\geq 0}$. This leads to a relationship to the

Optimum Communication Spanning Tree Problem, which is described in Subsection 3.4.3. Some of the notations are borrowed from [101].

Let $e = uv$ be an edge and T a spanning tree of a weighted graph and remember that $\mathrm{dist}_G(u,v)$ denotes the length of a shortest u-v-path in G. Then $\mathrm{Path}_T(u,v) = \mathrm{Path}_T(e)$ denotes the unique u-v-path in T, regarded as a set of edges. The *stretch sum* is defined as $\mathrm{StrSum}(T) = \sum_{uv\in E\setminus T} \mathrm{dist}_T(u,v)$. With $\mathrm{mult}_T(e) = w(e)\cdot|\mathrm{Path}_T(e)|$ we denote the *multiplicity* of an edge. Finally, define the *multiplicity sum* of a spanning tree as $\mathrm{MultSum}(T) = \sum_{e\in E\setminus T} \mathrm{mult}_T(e)$. See Figure 3.1 for an example.

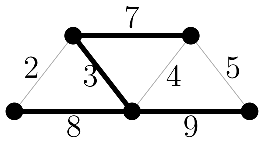


Figure 3.1: Graph with a highlighted spanning tree T, $\mathrm{StrSum}(T) = 40$ and $\mathrm{MultSum}(T) = 27$.

Taking a short look back to unweighted graphs, it turns out that the notions of stretch sum and multiplicity sum are equivalent. In contrast, this is not true for weighted graphs. However, when considering weighted planar graphs, there is another relationship of stretch sum and multiplicity sum.

In what follows, let $G = (V,E)$ be a planar graph, $G^* = (V^*, E^*)$ its dual graph, T a spanning tree of G, and T^* the dual tree of T in G^*. Concerning the weights of the dual graph, we simply extend the weight function $w : E \to \mathbb{R}_{\geq 0}$ of a planar graph $G = (V,E)$ to $w : E \cup E^* \to \mathbb{R}_{\geq 0}$ such that $w(\varphi_E(e)) = w(e)$ for all $e \in E$.

Lemma 3.18 ([101]). *Let $e_1, e_2 \in E$ be two edges in G and e_1^* and e_2^* the corresponding dual edges in G^*. Then $e_1 \in \mathrm{Path}_T(e_2)$ if and only if $e_2^* \in \mathrm{Path}_{T^*}(e_1^*)$.* □

Using our terminology and considering weighted graphs, Lemma 4.1 in [101] and its proof reads as below.

Theorem 3.19. *The stretch sum of a spanning tree T is equal to the multiplicity sum of its corresponding dual tree T^*, i.e. $\mathrm{StrSum}(T) = \mathrm{MultSum}(T^*)$.*

Proof. Directly by the definition of stretch sum, we get

$$\mathrm{StrSum}(T) = \sum_{e_1\in E\setminus T} \; \sum_{e_2\in \mathrm{Path}_T(e_1)} w(e_2).$$

We count for each edge $e_1 \in T$ the number of paths of edges not in T in which e_2 occurs, multiply this number with $w(e_2)$, sum up over all edges in T and achieve

$$\begin{aligned}
\mathrm{StrSum}(T) &= \sum_{e_2 \in T} \sum_{\substack{e_1 \in E \setminus T: \\ e_2 \in \mathrm{Path}_T(e_1)}} w(e_2) \\
&= \sum_{e_2 \in T} w(e_2) \cdot |\{e_1 \in E \setminus T \mid e_2 \in \mathrm{Path}_T(e_1)\}| \\
&\overset{\text{Lem. 3.18}}{=} \sum_{e_2^* \in E^* \setminus T^*} w(e_2^*) \cdot |\{e_1^* \mid e_1^* \in \mathrm{Path}_{T^*}(e_2^*)\}| \\
&= \sum_{e_2^* \in E^* \setminus T^*} w(e_2^*) \cdot |\mathrm{Path}_{T^*}(e_2^*)| \\
&= \mathrm{MultSum}(T^*).
\end{aligned}$$

□

With this notion in hand, for a strictly fundamental cycle basis in $G = (V, E)$ induced by the spanning tree T we achieve

$$\begin{aligned}
\Phi(T) &= \mathrm{StrSum}(T) + \sum_{e \in E \setminus T} w(e) \\
&\overset{\text{Thm. 3.19}}{=} \mathrm{MultSum}(T^*) + \sum_{e \in E \setminus T} w(e) \\
&= \mathrm{MultSum}(T^*) + \sum_{e^* \in T^*} w(e^*) \qquad (3.2) \\
&= \sum_{e^* \in E^*} w(e^*) \cdot |\mathrm{Path}_{T^*}(e^*)|. \qquad (3.3)
\end{aligned}$$

The equality in (3.2) holds due to the convention we made on the weights of the dual edges and by Definition 3.12.

3.4.3 Relationship to the Ocst Problem

We will now illustrate that minimizing the term on the left hand side of Equation (3.3) is a special case of the Optimum Communication Spanning Tree Problem, Ocst for short. Sometimes, e.g. in [42], this problem is also referred to as Minimum Communication Spanning Tree Problem. The optimization version of this problem reads as follows.

OCST		
Instance:	Complete graph	$G = (V, E)$,
	distance function	$d : E \to \mathbb{R}_{\geq 0}$,
	requirements	$r(u, v) \in \mathbb{R}_{\geq 0}$ for all $u, v \in V$.
Feasible solution:	Spanning tree	T.
Objective function:		$\sum_{u,v \in V} r(u, v)\mathrm{dist}_T(u, v)$.
opt:		min.

As usual, $\mathrm{dist}_T(u, v) := \sum_{e \in \mathrm{Path}_T(u,v)} d(e)$. In contrast to the literature (e.g. [31]) we use the term *distance function* to avoid confusions with two different weight functions in Equations 3.4.

The Optimum Communication Spanning Tree Problem is due to Hu who considered two special cases of this problem, see [61]. More precisely, he investigated the Optimum Requirement Spanning Tree Problem where $d \equiv 1$ on one hand, and on the other hand, the Optimum Distance Spanning Tree Problem (ODST) with $r(u, v) = 1$ for all $u, v \in V$ and a special form of triangle inequality for d. Both problems were shown to be solvable in polynomial time. Without the restriction of the distance function, the ODST is already $\mathcal{NP}$-complete (ND3 and ND7 in [53]).

The OCST itself is even $\mathcal{NP}$-complete if the (standard) triangle inequality does hold for d. At least, it is approximable within $\mathcal{O}(\log n \log \log n)$, in this case. If additionally all requirements are equal to 1, it even admits a PTAS, see [128]. In further lines of research, the problem was examined heuristically ([115]), and computational results were provided as well, see [109]. When the graph is not required to be complete, the OCST is approximable within a factor $\mathcal{O}(\log n \log \log n \log^3 \log \log n)$. This is even valid for arbitrary d ([1, 42]).

Remember that Hu studied *complete* graphs and all algorithmic, completeness, and approximability results mentioned above, excluding the last one, hold for complete graphs. In this regard, one has to study the references carefully since not all authors explicitly distinguish between the complete and the general case, see e.g. [2, 31, 42]. Nonetheless, let us remark that the OCST in its form given above—without any restrictions to the distance function or to the requirements—does not essentially differ from a version for arbitrary graphs. Therefore, the distance function is amplified to $d : E \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. This, in turn, can be avoided by choosing sufficiently large but still applicable values for d. To see the similarity between the OCST and a version for all graphs, let $G = (V, E')$ be a graph which is not necessary complete, and $d' : E' \to \mathbb{R}_{\geq 0}$ its distance function. The requirements $r(u, v)$ are still given for each pair of vertices. Observe that this problem is the same as the OCST on the complete graph $G = (V, E)$ with distance function d, by setting

$$d(e) = \begin{cases} d'(e), & e \in E' \\ \infty, & \text{otherwise.} \end{cases}$$

Similarly, minimizing (3.3) in a dual graph $G^* = (V^*, E^*)$ can be expressed as an Optimum Communication Spanning Tree Problem by setting

$$r(u^*, v^*) = \begin{cases} w(u^*v^*), & u^*v^* \in E^* \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad d(e^*) = \begin{cases} 1, & e^* \in E^* \\ \infty, & \text{otherwise.} \end{cases} \tag{3.4}$$

Note that we won't get into trouble with undefined terms like "$0 \cdot \infty$" because edges e^* with $d(e^*) = \infty$ are not chosen for the spanning tree. Otherwise, since all considered graphs are biconnected, the term in Equation (3.3) would be immediately infinite. Remark further that this is *not* the Optimum Distance Spanning Tree Problem since the Odst is defined on the complete graph with *all* distances being 1.

Unfortunately, the distance function in (3.4) is not metric. Thus, the approximation algorithm presented in [128] cannot be applied to approximate minimum strictly fundamental cycle bases within $\mathcal{O}(\log n \log \log n)$ on a weighted planar graph. However, maybe one can profit from the very simple structure of the distance function. On the other hand, we cannot even say that there is a PTAS for the unweighted and planar case since $r(u^*, v^*) = 1$ does not hold for all pairs of vertices in the completed graph.

A further possibility to attack the Ocst is to drop the completeness of the graph. Then, it can be investigated on miscellaneous graph classes, on which several structural properties can emerge as beneficial for treating the problem. As we will see in Section 3.7, the problem turns out to be tractable on outerplanar graphs—admittedly with further restrictions. To be accurate, the problem considered there reads as follows.

General Optimum Communication Spanning Tree (Gocst)		
Instance:	Arbitrary graph	$G = (V, E)$,
	requirements	$r(u, v) \in \mathbb{R}_{\geq 0}$ for all $uv \in E$.
Feasible solution:	Spanning tree	T.
Objective function:		$\sum_{uv \in E} r(u, v) \mathrm{dist}_T(u, v)$.
opt:		min.

Note that there is no distance function for this problem and $\mathrm{dist}_T(u, v)$ is simply the number of edges in the unique u-v-path in T. Moreover, the requirements are only defined for adjacent vertices. Thus, keeping these restrictions in mind, the term "general" is maybe somewhat overdrawn, although the problem is now defined on arbitrary graphs.

3.4.4 SFCBs on Non-Metric Planar Graphs

Clearly, if a graph is unweighted, many problems are more tractable than in the weighted case. If the graph provides a non-constant weight function, it is often helpfully when this function is metric, at least.

To get an impression, take a look at *average tree spanner problems*, cf. Chapter 3 in [129]. For a graph $G = (V, E)$ with a weight function w, a spanning tree T, and an edge $e = uv$, one can choose the `domain` from $\{E \setminus T, E, V \otimes V := V \times V \setminus \{(v, v) \mid v \in V\}\}$ and the `term` to sum over from $\{\mathrm{dist}_T(u, v), \mathrm{dist}_T(u, v) + w(e), \frac{\mathrm{dist}_T(u,v)}{w(e)}, \frac{\mathrm{dist}_T(u,v)}{\mathrm{dist}_G(u,v)}\}$. This leads to twelve combinations. Dropping the two combination which are not meaningful, namely which involve both $V \otimes V$ and $w(e)$, leaves ten. For an arbitrary weight function w only two pairs of problems coincide. In contrast, three of the remaining problems collapse to one single problem if the weight function is metric. Finally, if there are unique weights, there are essentially only three average tree spanner problems anymore.

In this subsection, we are concerned with cycle bases on planar graphs with non-metric weight functions. More precisely, we consider the following technical lemmas, where Lemma 3.21 restricts the number of basic circuits which contain a non-metric arc. Intuitively, a non-metric arc should be contained in only one basic circuit.

Lemma 3.20 (Lemma 12.16. in [78]). *For every minimum cycle basis B and every arc a there exists a shortest circuit C that contains a with $C \in B$.*

Proof. Assume that there was a minimum cycle basis B and an arc a such that each shortest circuit through a is not in B. Let C^* be such a circuit. In the support of C^*, there must be a basic circuit C which contains a. But then, $B \setminus \{C^*\} \cup \{C\}$ would be a cycle basis with a weight less than the weight of B. □

Lemma 3.21 (Cf. Open Problem 6 in the Survey). *Let $a = uv$ be a non-metric arc of a biconnected directed graph D. Then, for each minimum directed cycle basis B, there is exactly one circuit $C \in B$ with $a \in C$. This is also true for each minimum undirected cycle basis.*

Proof. According to Lemma 3.20, we can pick out a shortest circuit C^* through a from the minimum cycle basis B. This circuit consists of the arc a and the u-v-path P. Since a is not metric, it holds that $w(P) < w(a)$. Now let $C \in B$ be another basic circuit that contains a. Then the circuit $C' := C \setminus \{a\} \cup P$ has a weight smaller than the weight of C. This implies that $B \setminus \{C\} \cup \{C'\}$ is a cycle basis with a smaller weight than B. □

Open Problem 6 in the Survey additionally asks whether this is true for any other type of cycle bases. In our next example, we will show that this lemma is *not* true for strictly fundamental cycle bases or for 2-bases. We argue with undirected graphs and note that the results can be directly carried over to the directed case.

Example 3.22.
For both statements, we consider the planar graph in Figure 3.2. Of course, the graph can be made simple by subdividing parallel edges. The edge e_1 is considered as a non-metric edge, so it must not be subdivided.

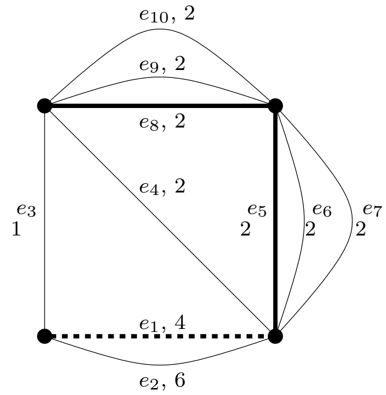


Figure 3.2: Planar non-metric graph with edges e_1 to e_{10}, indicated weight function, the non-metric edge e_1 (dashed) and a fundamental spanning tree T (fat edges).

A Minimum 2-Basis. The graph in Figure 3.2 is embedded in the plane such that the boundary of the unbounded face constitutes the longest plane circuit. The circuit $\{e_2, e_3, e_{10}, e_7\}$ is a plane circuit of maximum weight. Thus, according to Theorem 3.16 and the paragraph above, the boundaries of the bounded faces form a minimum 2-basis with a weight of 39. As one can see from the embedding, the non-metric edge e_1 is contained in two basic circuits.

A Minimum Strictly Fundamental Cycle Basis. The fundamental spanning tree T in Figure 3.2 induces a strictly fundamental cycle basis with a weight of 41. The non-metric edge e_1 is contained in two basic circuits. All other spanning trees which contain e_1 induce strictly fundamental cycle bases with greater weights. Anyway, for our purpose, it is sufficient only to consider the trees in which e_1 is a chord. We show that each such tree T with e_1 as a chord induces an SFCB with a weight of at least 42. Therefore, we distinguish four cases, based on the path in the tree between the end nodes of e_1.

Case 1 ($e_3, e_4 \in T$).
It immediately follows that $w(C_T(e_1)) = 7$ and $w(C_T(e_2)) = 9$. Additionally, one of the edges e_5 to e_{10} has to be in the tree. This induces two circuits of length 4 and three circuits of length 6. Altogether, we obtain a strictly fundamental cycle basis with a length of 42.

Case 2 ($e_2, e_4 \in T$).
At first, it is $w(C_T(e_1)) = 10$ and $w(C_T(e_3)) = 9$. Again, one of the edges e_5 to e_{10} has to be in the tree, inducing circuits with a summarized length of 26. The whole cycle basis has a length of 45.

Case 3 ($e_2 \in T$, $e_4 \notin T$).
In this case, the question arises how the common end vertex of the edges e_3 and e_4 is joined to the tree. Assume at first that $|\{e_5, e_6, e_7\} \cap T| = 1 = |\{e_8, e_9, e_{10}\} \cap T|$. Then the four chord edges in the two sets induce circuits with weight 16, $w(C_T(e_4)) = 6$, and $w(C_T(e_3)) = 11$. The combined weight of the SFCB is 43.

Now assume that either ($\{e_5, e_6, e_7\} \cap T = \emptyset$) or $\{e_8, e_9, e_{10}\} \cap T = \emptyset$. Then it holds that $e_3 \in T$. The cut $S_T(e_2)$ contains four edges of weight 2, each of it inducing a circuit of weight at least 9. Together with $w(C_T(e_1)) = 10$, this leads to an SFCB with a weight of at least 46.

Case 4 ($e_3 \in T$, $|\{e_5, e_6, e_7\} \cap T| = 1 = |\{e_8, e_9, e_{10}\} \cap T|$).
Analogously to the first part of Case 3, the four chord edges in the two sets mentioned above induce circuits with a weight of 16. Together with $w(C_T(e_1)) = 9$, $w(C_T(e_4)) = 6$, and $w(C_T(e_2)) = 11$, we get a strictly fundamental cycle basis with weight 42. $\Diamond$

Note that the 2-basis in the first paragraph of the example cannot be minimum among all cycle bases. This would be a contradiction to Lemma 3.21. Performing for example de Pina's MCB algorithm ([102], see also Algorithm 12.4 in [78]) results in the cycle basis $B = \{\{e_6, e_7\}, \{e_8, e_9\}, \{e_9, e_{10}\}, \{e_3, e_1, e_4\}, \{e_3, e_2, e_4\}, \{e_5, e_6\}, \{e_{10}, e_4, e_6\}\}$, which has the weight $\Phi(B) = 38$. Note that another ordering of the edges had been used to compute this basis. However, the non-metric edges e_1 and e_2 appear in only one circuit of this basis.

In this subsection, we showed that Lemma 3.21 does not hold for 2-bases or for strictly fundamental cycle bases. This solves—at least partially—Open Problem 6 in the Survey.

3.5 $\mathcal{NP}$-completeness of the MSFCB in Planar Graphs

This section is dedicated to establish the $\mathcal{NP}$-completeness of the MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS Problem on unweighted planar graphs. Similar to the proof presented in Section 2.3, also this proof will be by reduction from the planar version of the EXACT COVER BY 3-SETS Problem. Again, we profit from the embedding and the connection of the 3-sets, which has been described in Subsection 2.3.1 in detail. Subsection 3.5.1 is dedicated to the presentation of the discussed problems. In Subsection 3.5.2, the connection of the 3-sets is specified to the reduction in this section. Also the reduction itself is performed in Subsection 3.5.2.

3.5.1 The Problems

In this subsection, we fix the notations of the problems. The decision version of MSFCB restricted to planar graphs reads as follows.

P-MSFCB

Instance:	Planar graph $G_S = (V_S, E_S)$, positive integer k.
Question:	Does G_S have an SFCB B with $\Phi(B) \leq k$, or more precisely, does G_S have a spanning tree T with $\sum_{e \in E \setminus E(T)} C_T(e) \leq k$?

Lemma 3.23. *The* P-MSFCB *is in* $\mathcal{NP}$.

Proof. Given a fundamental spanning tree,the length of a fundamental circuit C can be determined by performing a depth first search on the tree edges from one end node of the private edge of C to the other one. This can be done in time $\mathcal{O}(n)$. Since there are $\nu \in \mathcal{O}(m)$ private edges, it is possible in time $\mathcal{O}(nm)$ to decide whether the size of a given strictly fundamental cycle basis is at most k. □

As already mentioned above, our proof will be by reduction from the planar version of the EXACT COVER BY 3-SETS Problem whose $\mathcal{NP}$-completeness has been established in [41]. This problem was already presented in Subsection 2.3.2, for the sake of completeness it is listed here again.

P-X3C	
Instance:	Integers n, m with $3\|n$, set $X = \{1, 2, \dots, n\}$, subset $\mathscr{S} \subset \mathcal{P}(X)$ with $\|\mathscr{S}\| = m$ and $\|S\| = 3$ for all $S \in \mathscr{S}$, where $G_X = (V_X, E_X)$ with $V_X = X \cup \mathscr{S}$ and $E_X = \bigcup_{S \in \mathscr{S}} \{\{S, x\} \mid x \in S\}$ is planar.
Question:	Is there a subset $\mathscr{S}' \subseteq \mathscr{S}$ such that for all $x \in X$ there is exactly one $S \in \mathscr{S}'$ with $x \in S$?

Remark 3.24. *The problem is only interesting for* $3 \leq n \leq 3m = |E_X|$.

3.5.2 The Transformation

Given an instance of P-X3C with an embedded graph G_X, we transform this graph to a planar graph G_S and compute a number k such that there is a fundamental spanning tree on G_S with $\Phi(T) \leq k$ iff there is a subset $\mathscr{S}' \subseteq \mathscr{S}$ as posed in the definition of P-X3C. As in Section 2.3, the nodes of G_X in $\mathscr{S}$ are called *set nodes*, while the nodes in X are referred to as *element nodes*.

The first step of the transformation is to use the ideas presented in Subsection 2.3.1, i.e. the construction of a spinal tree which connects the element nodes to each other. Here, this can be done in an easier way because there is no restriction to the degrees in the graph G_S, contrary to the graph G_O constructed in Subsection 2.3.3. Thus, simply connect the m set nodes in $\mathscr{S}$ with a set E_S of $m-1$ *spinal edges* in a way that there is a unique path consisting only of spinal edges between each two set nodes. Note that this step is not unique and that there can be faces which contain no or more than one spinal edge. Figure 3.3 illustrates this construction with an example with $X = \{1, 2, \dots, 6\}$ and $\mathscr{S} = \{\{2, 3, 5\}, \{1, 2, 6\}, \{1, 4, 6\}, \{1, 5, 6\}\}$. The element nodes are depicted as numbered circles, while the rectangles represent the set nodes.

The set nodes and the edges in E_X and E_S will be replaced with components described below. The components for the set nodes will get special *spinal nodes* on which the spinal

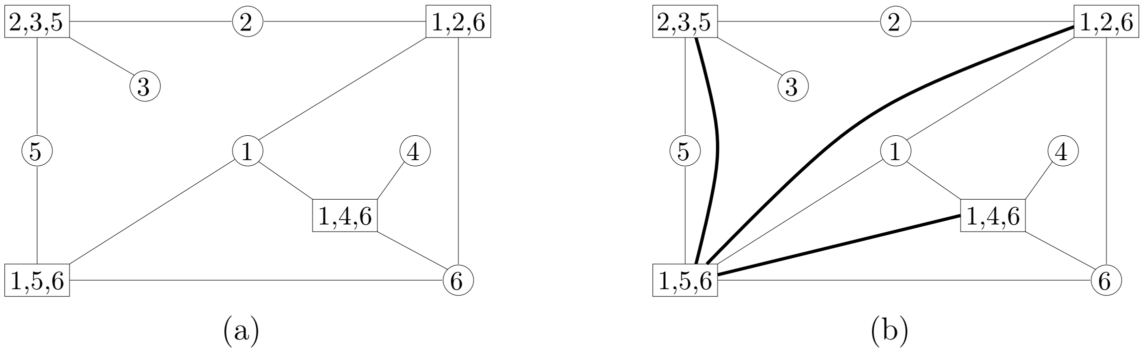


Figure 3.3: Embedded instance of the P-X3C Problem (a). The instance after the insertion of $m-1$ spinal edges of the spinal tree (b).

tree can be attached. In the next paragraphs, we describe these components and identify the necessary properties for the proof. Moreover, the integer k for the P-Msfcb instance is computed. Again, we distinguish between spanning trees which are *regular* and *irregular* on a component. Although this notion of regularity always corresponds to a spanning tree, we often call the component itself (ir)regular, for short. Directly after the description in each paragraph, we compute the part of the parameter k on the assumption that the tree is regular on the component. The corresponding part of k, where the tree is irregular, is computed below in several extra paragraphs, after the computation of the integer k itself. At the end of this subsection, we give a graphical overview of the results which lead to the main theorem, i.e. the $\mathcal{NP}$-completeness of the P-Msfcb.

Thick Bunches. Each of the $3m$ edges in E_X and each of the $m-1$ spinal edges is substituted by a *thick bunch*. A thick bunch arises out of an edge uv by replacing it with m^7 paths of length 2. The m^7 new nodes are denoted *center nodes*. There are essentially two possibilities of a spanning tree restricted to a thick bunch. When one center node has degree two in the spanning tree, the thick bunch is called *regular*. Otherwise, if all center nodes are leaves of the tree, the thick bunch is termed *irregular*. In what follows, regularity of several components is always associated with a spanning tree. See Figure 3.4 for a regular and an irregular thick bunch.

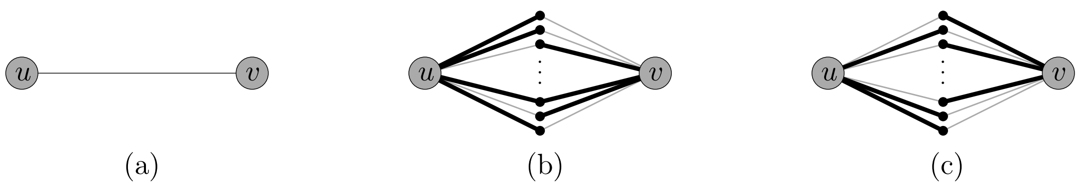


Figure 3.4: An edge uv (a), a regular (b) and an irregular (c) thick bunch. Tree edges are drawn as thick lines.

If all thick bunches are regular, the chords contained in the thick bunches induce circuits with total length of

$$\Phi^r_{\text{tb}} = (\underbrace{4m-1}_{\substack{\text{number of}\\ \text{thick bunches}}}) \cdot (\underbrace{m^7-1}_{\substack{\text{circuits per}\\ \text{thick bunch}}}) \cdot \underbrace{4}_{\substack{\text{length of}\\ \text{one circuit}}} . \tag{3.5}$$

Long Bunches. *Long bunches* are parts of gears, which are described later. Gears will replace the set nodes in $\mathscr{S}$, and each gear will contain six long bunches. Hence, the graph G_S will contain $6m$ long bunches. A long bunch is constructed by subdividing an edge uv with two additional nodes and by replacing each of the three originated edges with a *slice* of m^5 paths with lengths $6m^2$. If all edges in a path are in the tree, the path is called an *complete path*. Otherwise, i.e. one edge of the path is missig in the tree, it is referred to as *incomplete path*. The terms *regular* and *irregular* are similarly used as in the context of thick bunches. See also Figure 3.5.

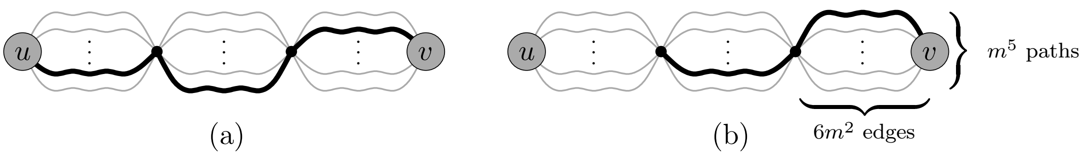


Figure 3.5: A regular (a) resp. irregular (b) long bunch. Every edge of a fat path is contained in the tree while exactly one edge of a grey path is missing in the tree.

Again, we assume that all long bunches are regular for the moment. Then, the chords contained in the long bunches induce circuits of length

$$\Phi^r_{\text{lb}} = \underbrace{6m}_{\substack{\text{number of}\\ \text{long bunches}}} \cdot \underbrace{3}_{\substack{\text{slices per}\\ \text{long bunch}}} \cdot \underbrace{(m^5-1)}_{\substack{\text{circuits}\\ \text{per slice}}} \cdot \underbrace{2 \cdot 6m^2}_{\substack{\text{length of}\\ \text{one circuit}}} . \tag{3.6}$$

Switchers. These components help to control whether the gears which model the 3-sets belong to $\mathscr{S}'$ or $\mathscr{S} \setminus \mathscr{S}'$. See Figure 3.6 for an illustration.

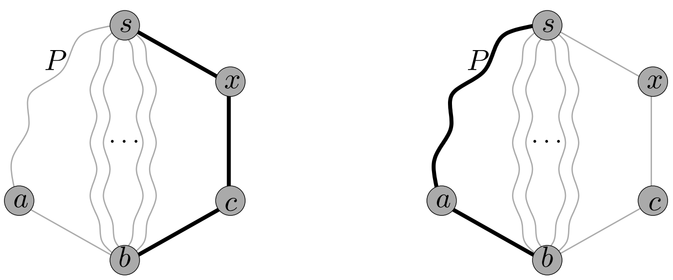


Figure 3.6: Active (left) and inactive switcher (right).

The a-s-path P has a length of $5m$ edges. In the interior of a switcher are $130m^3$ *interior b-s-paths*, each of length $16m^2$. Contrary to thick or long bunches, there are basically two types of regular switchers. A switcher is *regular* if exactly one of the conditions below is fulfilled.

- The edges sx, cx and bc are contained in the corresponding tree. In this case, the switcher is referred to as *active*.
- The spanning tree contains the complete path P and the edge ab. These regular switchers are called *inactive*.

All other switchers are *irregular*, i.e. one of the $130m^3$ interior b-s-paths is completely in the tree or the unique b-s-path in the tree uses edges outside of the switcher. Note that an active switcher is allowed to contain the complete path P or the edge ab. Similarly, an inactive switcher can contain one or two of the edges sx, cx or bc. Anyway, we will see that all these cases are not possible if the switcher is assembled in a gear whose long bunches are all regular. The terms *complete* and *incomplete path* are used equivalently to the case of long bunches.

The estimation of the combined circuit length induced by chords in regular switchers is caught up on in the next paragraph, because the circuits contain also edges outside of the regular switcher.

Gears. A *gear* G^S substitutes a set node $S \in \mathscr{S}$. One gear consists of three switchers and six long bunches which are arranged as shown in Figure 3.7. A gear is called *regular* if all six long bunches and all three switchers are regular.

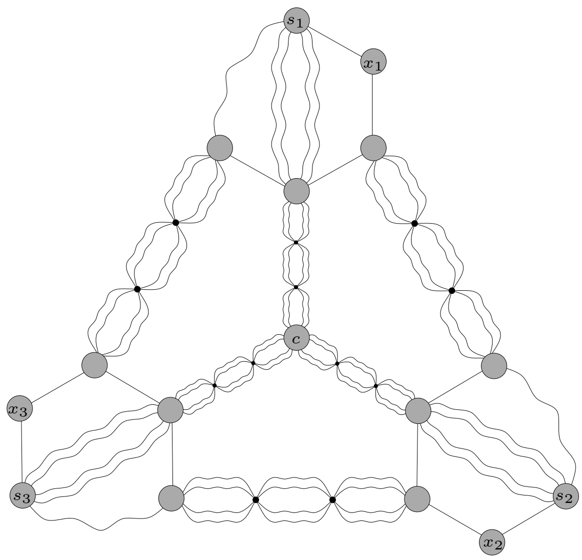


Figure 3.7: Arrangement of the three switchers and the six long bunches in a gear.

The X-*nodes* x_1, x_2 and x_3 in the gear G^S are connected via thick bunches to the three element nodes which are in the 3-set S. The *spinal nodes* s_1, s_2 and s_3 can be used to connect the gears to each other with thick bunches which replace the spinal edges of the spinal tree introduced in the first step of the transformation. Clearly, not all spinal nodes will be used therefore, because there are $3m$ spinal nodes but only $m-1$ spinal edges which connect them. Figure 3.8 illustrates how all components are assembled with a complete example.

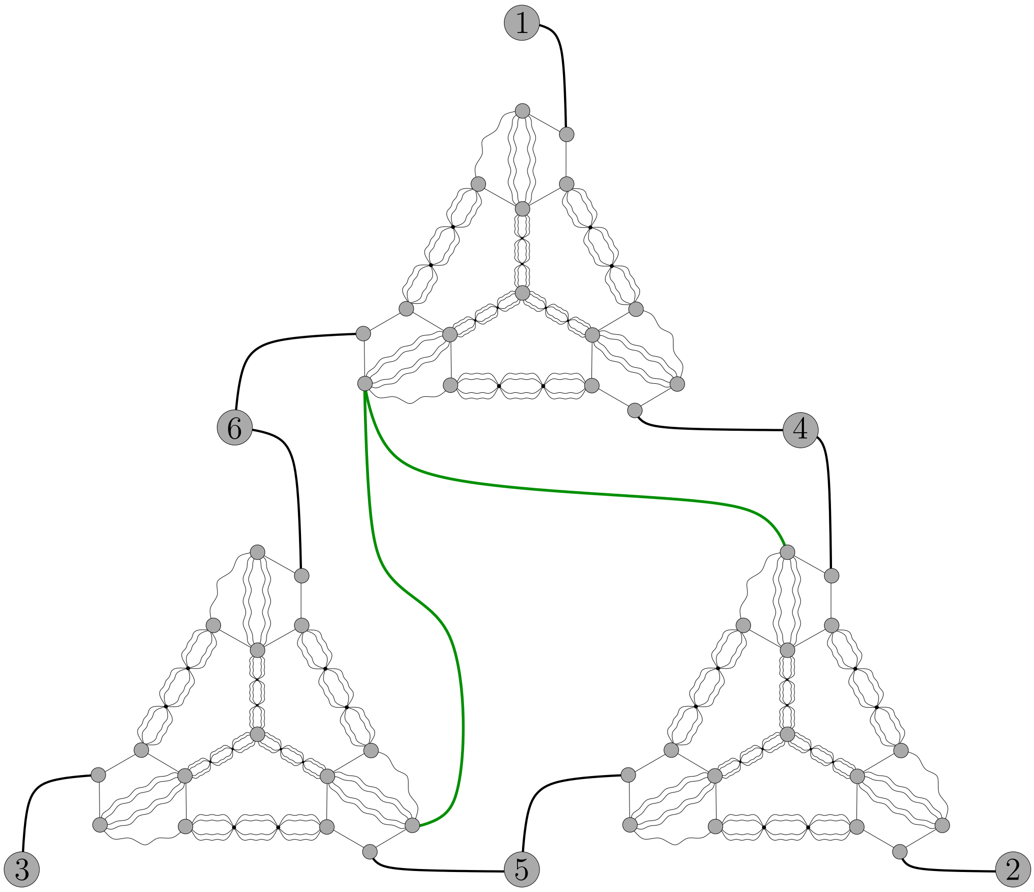


Figure 3.8: Complete example for the instance $X = \{1,2,3,4,5,6\}$, $\mathscr{S} = \{\{1,4,6\},\{3,5,6\},\{2,4,6\}\}$. Thick bunches which replaced the edges in E_X are drawn as thick black lines. The fat green lines represent thick bunches which replaced the spinal edges.

Lemma 3.25. *Let G_R be an embedding of the graph G_X as in the description of* P-X3C *where the nodes in $\mathscr{S}$ are replaced by gears. G_R shall not contain the spinal tree, yet. Then for each gear, the boundary of every adjacent face contains at least one of the gear's spinal nodes.* □

Proof. This is due to the fact that X-nodes and spinal nodes of one gear G^S alternate in the boundary of the unbounded face of G^S. □

It follows that a transformation described in Subsection 2.3.1 and specified in Subsection 3.5.2 is actually possible while preserving planarity. Note that it is also feasible that one spinal node can be adjacent to no or to more than one thick bunch.

Proposition 3.26. *The constructed graph is planar. Moreover, the transformation can be done in polynomial time.* □

The following observation eliminates further cases in the estimation of the circuit lengths induced by chords in regular switchers.

Observation 3.27. *In a regular gear, the switchers have the same orientation, i.e. all three switchers are either active or all three are inactive. With Figures 3.6 and 3.7 in mind, if all three switchers are active then edge ab and one edge of P is missing in the tree. On the other hand, when the switchers are inactive edge bc and at least one of cx and sx are chords. As we will see later in Corollary 3.33, actually all three edges have to be chords.* □

Otherwise, if there would be an active *and* an inactive switcher in a regular gear, T would contain a circuit and it was not connected. If all three switchers of a regular gear are (in)active, the gear itself is also denoted *(in)active.*

Proposition 3.28. *If all thick bunches, all long bunches, and all switchers are regular then at most n switchers can be active.* □

Otherwise, there would be an $x \in X$ which is connected to two active switchers. The corresponding gears are also connected to each other via the replaced spinal edges, and hence, T would contain a circuit. On the other hand, it is possible that less than n switchers are active. An element in X could also be connected to an inactive switcher if the edges cx or sx are in the tree. Lemma 3.32 will ensure that the path P avoids too many inactive switchers.

Lemma 3.29. *Let $\mathscr{S}$ and X be an instance of* P-X3C *and G_S a transformed graph in which all thick bunches, all long bunches, and all switchers are regular, and exactly n switchers are active. Then there is an exact cover for the* P-X3C *instance if and only if the constructed subgraph is a tree.*

Proof. Let $\mathscr{S}'$ be an exact cover. Set all switchers in gears corresponding to $\mathscr{S}'$ as active. Then each $x \in X$ is connected to exactly one gear via an active switcher. Since the gears are connected via the thick bunches which replaced the spinal edges, the constructed subgraph is a spanning tree.

Now assume that there is no solution of the P-X3C instance. Then for every choice $\mathscr{S}' \subseteq \mathscr{S}$ there is an $x \in X$ with

$$|\{S \in \mathscr{S}' \mid x \in S\}| \;=\; 0 \quad \text{or} \tag{3.7}$$

$$|\{S \in \mathscr{S}' \mid x \in S\}| \;\geq\; 2. \tag{3.8}$$

We identify the possibilities to set the switchers of the gears with the choice $\mathscr{S}'$, i.e. $S \in \mathscr{S}'$ if and only if the switchers of the corresponding gear are active. Then x is not connected to the rest of the graph (3.7) or it induces a circuit since it is connected to at least two gears (3.8), which are also connected with spinal edges. Thus, the constructed subgraph is not a spanning tree. □

Now, we are able to determine the size of the circuits induced by chords in active switchers. Therefore, assume that all thick and long bunches are regular. Figure 3.9 shows the setting for one active switcher in a regular gear. We describe the circuits which are closed by chords in the lower switcher. The color of the terms below the braces corresponds to the part of the gear in Figure 3.9. The chords induce circuits with a total length of

$$\Phi^r_{\text{as}} = \underbrace{n}_{\substack{\text{number of}\\ \text{active switchers}}} \cdot \Big[\underbrace{130m^3}_{\substack{\text{number of}\\ \text{int. paths}}} \cdot (\underbrace{16m^2+3}_{\substack{\text{int. path lengths}\\ \text{and } e_2,\, e_3,\, e_4}}) \tag{3.9}$$

$$+ \ (\overbrace{\underbrace{2}_{e_1,e_5} + \underbrace{3\cdot 3\cdot 6m^2}_{\text{3 long bunches}}}^{e_1}) \tag{3.10}$$

$$+ \ (\overbrace{\underbrace{5m}_{P} + \underbrace{3\cdot 3\cdot 6m^2}_{\text{3 long bunches}} + \underbrace{4}_{\substack{e_2,e_3,\\ e_4,e_5}}}^{e_P})\Big]. \tag{3.11}$$

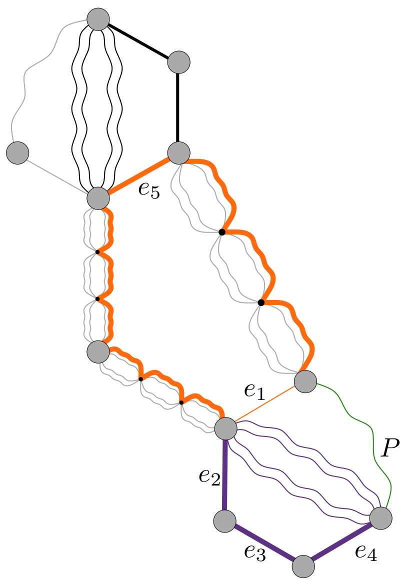


Figure 3.9: Circuits closed by chords in an active switcher. Tree edges and complete paths are depicted as thick lines, chords and incomplete paths as thin lines. The colors correspond to the terms in the equation above (Lines 3.9 to 3.11).

The first summand in the squared brackets in Line (3.9) is the total size of the circuits closed by the $130m^3$ interior paths and the tree edges e_2, e_3 and e_4. The chord e_1 in the lower switcher induces a circuit with three long bunches and the edge e_5, see Line (3.10). There is one chord e_P in the path P which closes a circuit with the three long bunches, e_5 in the upper switcher, and the edges e_2 to e_4. This is expressed in Line (3.11).

Now let us turn our focus to inactive switchers. In accordance with Lemma 3.29, there are exactly $3m - n$ inactive switchers if and only if there is an exact cover. Contrary to active switchers, we are now only able to estimate the circuit lengths. Therefore, consider the switcher in the upper left corner of Figure 3.10. The total size of circuits induced by chords in inactive switchers is at most

$$
\begin{aligned}
\Phi^r_{\text{is}} \leq \underbrace{(3m-n)}_{\substack{\text{number of}\\ \text{inactive switchers}}} \cdot \Big[& \underbrace{130m^3}_{\substack{\text{number of}\\ \text{int. paths}}} \cdot (\underbrace{16m^2 + 5m + 1}_{\substack{\text{int. path lengths,}\\ P_1 \text{ and } e_1}}) + (\overbrace{\underbrace{2}_{e_2, e_5} + \underbrace{3 \cdot 3 \cdot 6m^2}_{\text{3 long bunches}}}^{e_2}) && (3.12)\\
+ \underbrace{2}_{e_3, e_4} \cdot \Big(& \underbrace{4}_{\substack{\text{via } x\\ \text{to } x_1}} + (\underbrace{5 + 2 \cdot 3 \cdot 6m^2}_{\substack{\text{crossing the}\\ \text{active gear } G}}) && (3.13)\\
+ & \overbrace{(n-1)(\underbrace{2 \cdot 3 \cdot 6m^2 + 6}_{\substack{\text{crossing of an}\\ \text{active gear}}} + \underbrace{2}_{\substack{\text{thick bunch to}\\ \text{the next gear}}})}^{\text{crossing at most } n-1 \text{ further active gears}} && (3.14)\\
+ & \overbrace{(3m-n)(\underbrace{2 \cdot 3 \cdot 6m^2 + 2 \cdot (5m+1)}_{\substack{\text{crossing of an}\\ \text{inactive gear}}} + \underbrace{2}_{\substack{\text{thick bunch to}\\ \text{the next gear}}})}^{\text{crossing at most } 3m-n \text{ inactive gears}} \Big) \Big] && (3.15)
\end{aligned}
$$

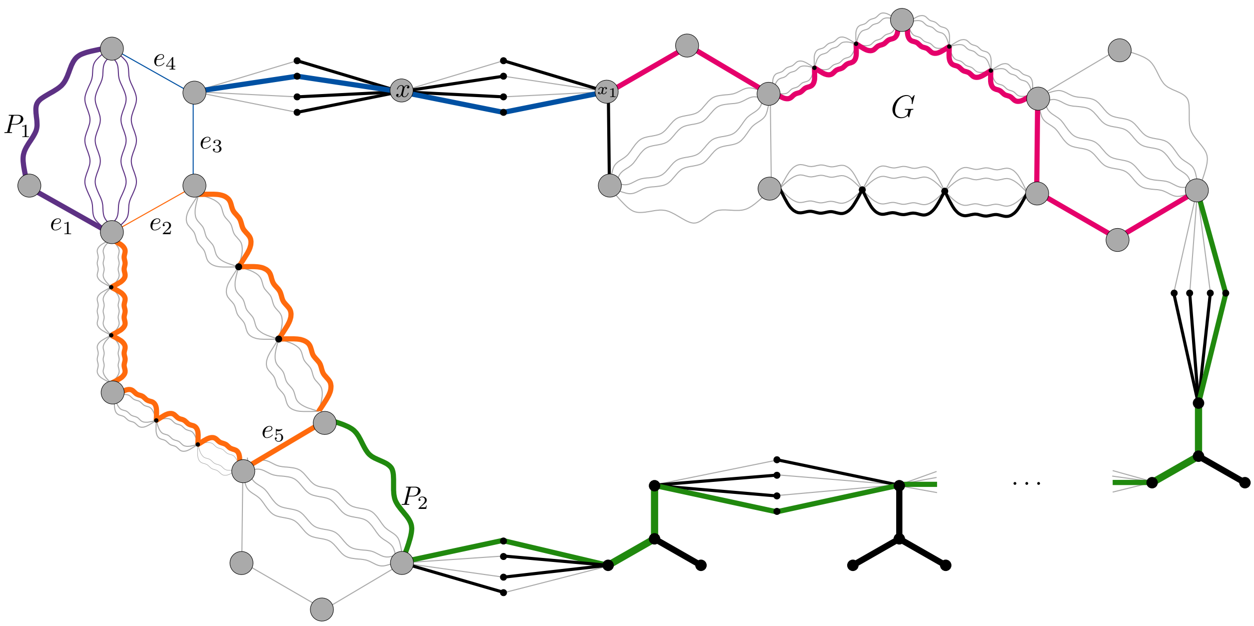


Figure 3.10: Circuits induced by chords in an inactive switcher.

Similar to the switcher in Figure 3.9, the switcher in the upper left corner of Figure 3.10 contains $130m^3$ interior paths. But here, they induce circuits of lengths $16m^2+5m+1$. The chord e_2 closes a circuit of the same length as e_1 in Figure 3.9. Lines (3.13) to (3.15) are dedicated to estimate the circuit lengths induced by e_3 and e_4. This is done simultaneously for both edges. At first, the circuits pass node x and meet the X-node x_1, indicated by blue edges. Then, they cross the active gear G, highlighted in magenta. Afterwards, the circuits have to cross at most $n-1$ further active and at most $3m-n$ inactive gears via green edges. Both, active and inactive gears are indicated in Figure 3.10 by very fat drawn $K_{3,1}$'s. Note that the last inactive gear is the gear which contains the edges e_3 and e_4.

The Integer k. After the construction of the graph G_S, we go on with the other part of the P-MSFCB instance, namely the integer k. Its value is the sum

$$\begin{aligned} k &:= \Phi^r_{\text{tb}} + \Phi^r_{\text{lb}} + \Phi^r_{\text{as}} + \Phi^r_{\text{is}} && (3.16)\\ &= 232m^8 - 4m^7 + 6240m^6 + 1950m^5 + (1038 - 650n)m^4 && (3.17)\\ &+ (126 + 44n)m^3 + (72 - 66n)m^2 + (20n^2 + 5n - 4)m + (4 + 2n - 8n^2). && (3.18) \end{aligned}$$

In this polynomial, only the coefficients in Line (3.17) will be of interest. To yield a strictly fundamental cycle basis with a size of at most k, it is necessary that all components—thick bunches, long bunches, and switchers—are regular. Furthermore, exactly $3m-n$ switchers may be inactive. At next, we will show that the strictly fundamental cycle basis has a size greater than k if the inducing spanning tree is irregular on any component or if more than $3m-n$ switchers are inactive.

Irregular Thick and Long Bunches. Assume at first that p_{tb} thick bunches are irregular. Since G_X was bipartite, each chord in an irregular thick bunch induces a circuit with length of at least 5. Thus, the chords of all thick bunches together close circuits with total length

$$\Phi^i_{\text{tb}} \geq (4m - 1 - p_{\text{tb}}) \cdot (m^7 - 1) \cdot 4 + p_{\text{tb}} \cdot m^7 \cdot 5. \tag{3.19}$$

Now let p_{lb} long bunches be irregular. In an irregular long bunch, two of the three slices look like in a regular long bunch. This is regarded by the first summand inside of the squared brackets in Line (3.21). In the remaining slice, all m^5 chords induce circuits with lengths of at least three times the path length of a path in a slice, i.e. $3 \cdot 6m^2$. This is considered by the second summand in Line (3.21). Thus, the circuit lengths of chords in long bunches sum up to

$$\begin{aligned} \Phi^i_{\text{lb}} &\geq (6m - p_{\text{lb}}) \cdot 3 \cdot (m^5 - 1) \cdot 2 \cdot 6m^2 && (3.20)\\ &+ p_{\text{lb}} \cdot \left[2 \cdot (m^5 - 1) \cdot 2 \cdot 6m^2 + m^5 \cdot 3 \cdot 6m^2\right]. && (3.21) \end{aligned}$$

Thick and long bunches are now viewed simultaneously. Then, after adding Φ^i_{tb} and Φ^i_{lb}, the cycle basis has the size

$$\Phi > \Phi^i_{\text{tb}} + \Phi^i_{\text{lb}} \geq 232m^8 + (p_{\text{tb}} + 6p_{\text{lb}} - 4)m^7 + o(m^7). \tag{3.22}$$

Looking now at the coefficient of m^7 and compare it with its counterpart in Line (3.17), it can only be at most -4 if $p_{\mathrm{tb}} = p_{\mathrm{lb}} = 0$. It follows

Lemma 3.30. *An SFCB can only have a value of at most k if all thick and all long bunches are regular.* □

Irregular Switchers. As already mentioned after the explanation of regular switchers, there are essentially two types of irregular switchers. Thus, let p_{ip} switchers contain a complete interior path and let p_{np} be the number of further irregular switchers without such a path. The remaining $3m - p_{\mathrm{ip}} - p_{\mathrm{np}}$ switchers can be regarded as active. Allthough this is in general not possible by Proposition 3.28, it is sufficient for the estimation and makes it less involved.

In a switcher with one complete interior path P in the tree, the remaining $130m^3 - 1$ paths close circuits with P, their total length is

$$\Phi^i_{\mathrm{ip}} = p_{\mathrm{ip}} \cdot (130m^3 - 1) \cdot 2 \cdot 16m^2. \tag{3.23}$$

Now, take a look at Figure 3.6 and assume that the unique b-s-path in the tree contains edges outside of the switcher. Then, the $130m^3$ circuits induced by the interior paths contain at least one of the three long bunches starting at a, b or c. These circuits have a total length of

$$\Phi^i_{\mathrm{np}} \geq p_{\mathrm{np}} \cdot 130m^3 \cdot (16m^2 + 3 \cdot 6m^2). \tag{3.24}$$

For the remaining regular and active switchers, the interior paths close circuits with total length of exactly

$$\Phi^r_{\mathrm{s}} = (3m - p_{\mathrm{ip}} - p_{\mathrm{np}}) \cdot 130m^3 \cdot (16m^2 + 3). \tag{3.25}$$

With the assumption that all thick and long bunches are regular, we obtain a cycle basis of length

$$\begin{aligned} \Phi &> \Phi^r_{\mathrm{tb}} + \Phi^r_{\mathrm{lb}} + \Phi^i_{\mathrm{ip}} + \Phi^i_{\mathrm{np}} + \Phi^r_{\mathrm{s}} & (3.26)\\ &\geq 232m^8 - 4m^7 + 6240m^6 + (2080p_{\mathrm{ip}} + 2340p_{\mathrm{np}})m^5 + o(m^5). & (3.27) \end{aligned}$$

Analogously to the last paragraph, look at the coefficient of m^5 and compare it with its counterpart in Line (3.17). This coefficient can only be 1950 or less if $p_{\mathrm{ip}} = p_{\mathrm{np}} = 0$. Thus, we can strengthen Lemma 3.30 to

Lemma 3.31. *A strictly fundamental cycle basis on G_S can only have a size of at most k if all thick bunches, all long bunches, and all switchers are regular.* □

Note that Lemma 3.31 does only hold for sufficiently large m, since several coefficients of the lower terms in Equations (3.22) and (3.27) are actually smaller than the corresponding coefficients in Lines (3.17) and (3.18).

From Proposition 3.28 it is known that at most n switchers can be active if all thick and long bunches, as well as all switchers are regular. We now go on to show that n active switchers are necessary to obtain an SFCB of size k or less. Therefore, assume that $3m-n+p_{\text{is}}$ are inactive and $n-p_{\text{is}}$ are active. Bearing the regular thick and long bunches in mind, one obtains

$$\begin{aligned}
\Phi \;&>\; \Phi^r_{\text{tb}} + \Phi^r_{\text{lb}} + (3m-n+p_{\text{is}}) \cdot 130m^3 \cdot (16m^2+5m+1) && (3.28)\\
&\quad +(n-p_{\text{is}}) \cdot 130m^3 \cdot (16m^2+3) && (3.29)\\
&\geq\; 232m^8 - 4m^7 + 6240m^6 + 1950m^5 + (390-650n+650p_{\text{is}})m^4 + o(m^4). && (3.30)
\end{aligned}$$

Again, we compare the coefficients of m^4 in Lines (3.30) and (3.17). Obviously, it holds $(390-650n+650p_{\text{is}}) \leq (1038-650n)$ iff $p_{\text{is}}=0$. Hence, additionally to the regularity of all bunches and switchers, an SFCB of size k or less can only be achieved if exactly n switchers are active. Together with Observation 3.27 it follows

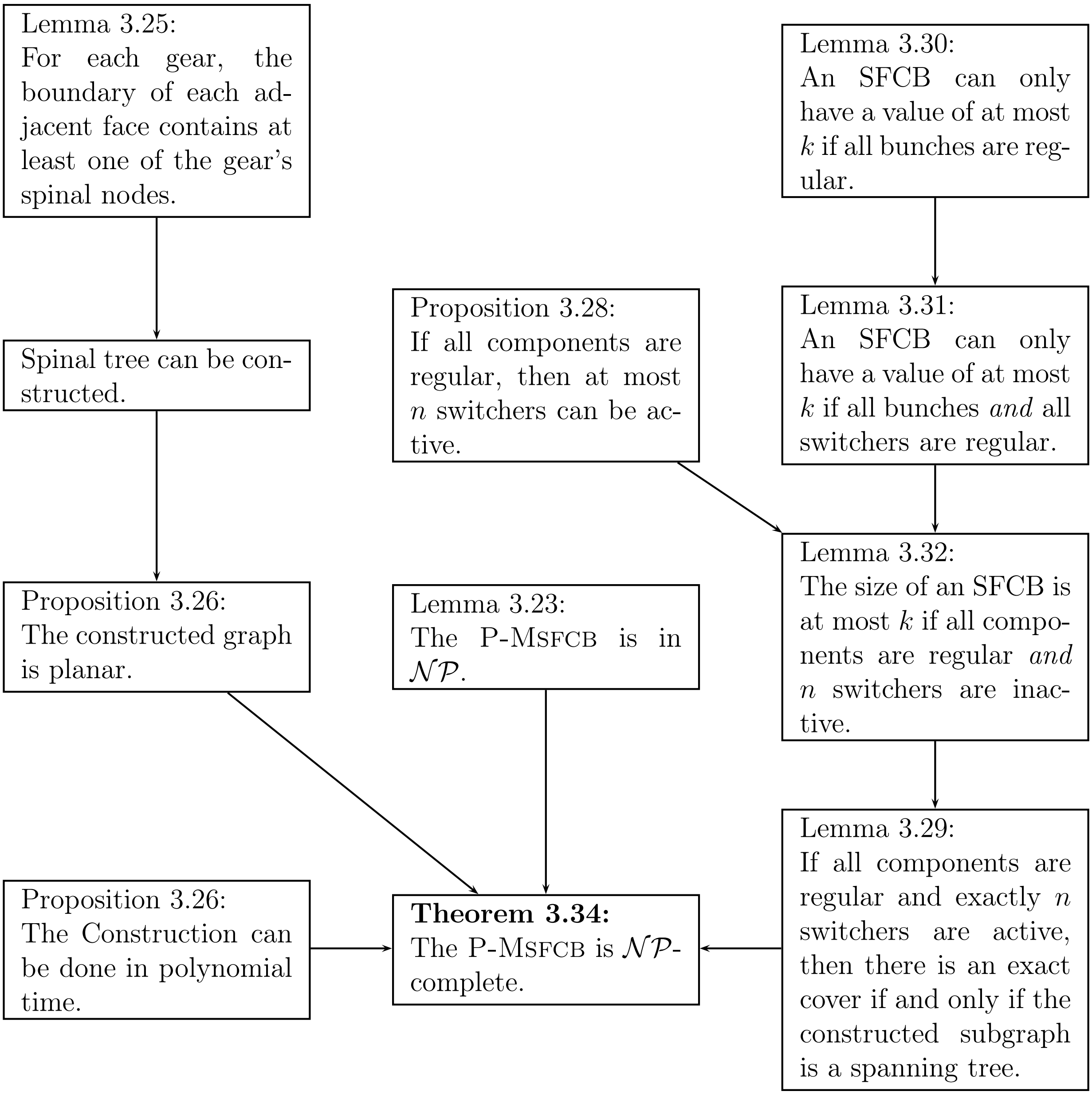
Lemma 3.32. *An SFCB on G_S can only attain a size of at most k if—in addition to the regularity of all bunches and switchers—n switchers are active.* □

Corollary 3.33. *Lemma 3.32 ensures that an element $x \in X$ cannot be connected to an inactive switcher by using one of the edges sx and cx, cf. Figure 3.6.*

We are now at a point at which we can summarize the statements in this subsection. We introduced several graph components and derived an integer k. We verified that all components have to be regular if an SFCB on the transformed graph shall not exceed k (Lemma 3.31). In Lemma 3.32, it is stated that additionally exactly n switchers have to be active. Putting all these regular components together results in a spanning subgraph. Lemma 3.29 points out that this subgraph is a spanning tree if and only if there is an exact cover on the P-X3C instance. Proposition 3.26 reveals the preservation of planarity and states that the construction can be done in polynomial time. Finally, Lemma 3.23 ensures the membership of the P-MSFCB in $\mathcal{NP}$. All these statements together lead to

Theorem 3.34. *The* MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS *Problem is $\mathcal{NP}$-complete on planar graphs.* □

The diagram below summarizes how Theorem 3.34 is derived from the statements in this subsection.



After the treatment of the element nodes which we described in Subsection 2.3.4 and according to Remark 2.6, Theorem 3.34 can be extended to

Theorem 3.35. *The* MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS *Problem is* $\mathcal{NP}$-*complete on biconnected planar graphs.* $\square$

3.6 Outerplanar Graphs

The computation of a minimum strictly fundamental cycle basis of an unweighted outerplanar graph can be done in linear time via a breath first search in the dual graph, see [74]. Also the size of a minimum SFCB is linearly bounded ([108]). However, both results do not apply if the outerplanar graph is weighted. In this section, we further restrict outerplanar graphs to a class, on which the problem of finding a minimum SFCB can be solved in linear time. Additionally, the minimum size of an SFCB is *minor monotone*, an important attribute in the proof for the a priori upper bound on the minimum SFCB size.

After a short introduction with a historical outline on outerplanar graphs in Subsection 3.6.1, we define them in a stringent manner in Subsection 3.6.2. Subsection 3.6.3 is dedicated to the concept of minor monotonicity.

3.6.1 Introduction

Since their seemingly first appearance in 1967 ([25]), outerplanar graphs have constituted an important subclass of planar graphs. Informally, an outerplanar graph can be drawn into the plane such that each *vertex* lies in the boundary of one single face. This face is usually drawn as the unbounded face. In a certain sense, this property makes outerplanar graphs to a generalization of trees since for trees, each *edge* borders on one single face. Because outerplanar graphs are a very restricted graph class, one may have some hope that some problems are easier to solve on them in comparison to, say, planar graphs. Also, several graph parameters are bounded on outerplanar graphs. For example, the treewidth and the size of a (balancing) node separator is 2 at most in both cases ([58, 71]). In contrast, both parameters can have values in $\Theta(\sqrt{n})$ on planar graphs, see e.g. [5, 83]. In this context, it seems surprising that if a planar graph does not contain an induced cycle of even length, then its treewidth is at most 49 ([117]).

In practise, outerplanar graphs are used in a wide range of applications, for example in the research of the secondary structure of biopolymers, [74, 122]. However, due to the chemical properties of nucleic acids, the outerplanar graphs considered there have a maximum degree of 3. But nevertheless, most elements in the NCI database ([93]), which is a well-known database for small biomolecules provided by the National Cancer Institute, can be described as outerplanar graphs, cf. [60, 105]. Several other ranges of practical applications for outerplanar graphs are the field of telecommunication ([69]), the theory of electrical circuits ([94]) and, more recently, data mining ([130]). Additionally, outerplanar graphs play an important role for shortest path computation in planar graphs, see [47, 48].

The class of outerplanar graphs is also interesting from a theoretical point of view. In [106], they are used to study the number of general dissections of a polygon. In another line of research, it turned out to be surprisingly difficult to prove the following partition property for planar graphs: *every planar graph has an edge partition into two outerplanar* *graphs*. Actually, this conjecture posed in 1971 in [24] could not be proved before 2005, and not after several erroneous proofs and partial results, cf. [56].

3.6.2 Definition and Properties on Outerplanar Graphs

We start with a formal definition of outerplanar and maximal outerplanar graphs. Moreover, this subsection contains several general statements on outerplanar graph and some results on minimum strictly fundamental cycle bases on unweighted outerplanar graphs.

Definition 3.36 ((maximal) outerplanar). *A graph is an* outerplanar graph *if it is isomorphic to a plane graph $G = (V, E)$ with $V \setminus \partial f^\infty = \emptyset$, where f^∞ denotes the unique unbounded face, again. An outerplanar graph is* maximal outerplanar *if it is not outerplanar for each additional edge.*

The following two statements describe some elementary but nonetheless important properties of outerplanar graphs. The first one is a straightforward calculation.

Proposition 3.37. *A maximal outerplanar graph with n vertices has $m = 2n - 3$ edges and $n - 2$ bounded faces.* □

Lemma 3.38 (Lemmas 5. and 6. in [74]). *An outerplanar graph contains a Hamiltonian circuit if and only if it is biconnected. In this case, the Hamiltonian circuit is unique.* □

The edges contained in the Hamiltonian circuit are referred to as *Hamiltonian edges*, the other ones as *chord edges* of the outerplanar graph. If we demand $v \in \partial(f^\infty)$ for all v in an isomorphic plane graph, a biconnected outerplanar graph is uniquely embeddable. This follows from Theorem 3.10 and from the fact that placing an additional vertex into f^∞ and connecting it to all other vertices results in a 3-connected planar graph. If $v \in \partial(f^\infty)$ does hold for all v, the embedding is called an *outerplanar embedding*. Clearly, this embedding is unique if the outerplanar graph is biconnected.

Beyond the outerplanar embedding, an outerplanar graph provides much more embeddings. To be more precise, every circuit can appear as the boundary of f^∞. This is equivalent to the next lemma. Note that Lemma 3.39 is not true for general planar graphs.

Lemma 3.39. *In an outerplanar graph, each circuit is a plane circuit.*

Proof. As a rough idea of the proof, we take an outerplanar embedding of the graph and an arbitrary circuit. Then, roughly speaking, everything that is outside of this circuit, is folded into the inside. Hence, the given circuit becomes the boundary of the unbounded face, and this is a plane circuit. It is sufficient to prove the statement for biconnected outerplanar graphs, because a circuit can only appear in one single biconnected component.

More precisely, let $G = (V, E)$ be a biconnected outerplanar graph. The vertices are numbered along the unique Hamiltonian circuit, i.e. $V = \{v_0, v_1, \ldots, v_{n-1}\}$. Clearly, the

Hamiltonian circuit is a plane circuit. All other circuits contain a chord edge. Thus, picking an arbitrary circuit C, it consists of ℓ single chord edges which alternate with ℓ sequences of succeeding Hamiltonian edges. Such a sequence is also allowed to be empty. For the sake of clarity of the various indices, take a look at the example in Figure 3.11 (a). W.l.o.g. let $v_{i_{\ell-1}+j_{\ell-1}}v_0$ be a chord edge for suited indices which are determined later (dashed in Figure 3.11 (a)). For $0 \le k \le \ell$, denote $s_k := (v_{i_k}v_{i_k+1}, \dots, v_{i_k+j_k-1}v_{i_k+j_k})$ a sequence of succeeding Hamiltonian edges. Let $\circ$ denote the concatenation of sequences, then the circuit has the form $C = s_0 \circ (v_{i_0+j_0}v_{i_1}) \circ \dots \circ s_{\ell-1} \circ (v_{i_{\ell-1}+j_{\ell-1}}v_0)$, where $v_{i_{\ell-1}+j_{\ell-1}} = v_{n-1}$ and $i_0 \le i_0 + j_0 < i_1 \le \dots < i_{\ell-1} \le i_{\ell-1} + j_{\ell-1}$.

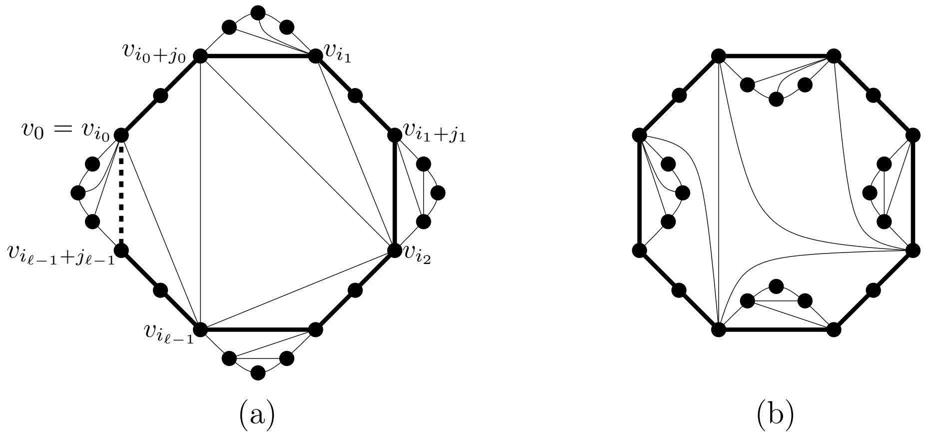


Figure 3.11: An outerplanar graph with a circuit C drawn with fat edges (a) and an embedding of this graph for which C is the boundary of the unbounded face (b).

Consider the induced subgraphs $G[\bigcup_{h=i_k+j_k}^{i_{k+1}} v_h]$ for $0 \le h \le \ell - 2$ and $G[\{v_{i_{\ell-1}+j_{\ell-1}}\} \cup \dots \cup v_{n-1} \cup v_0]$. Since the removal of the end nodes of the edges $v_{i_k+j_k}v_{i_{k+1}}$ and of the edge $v_{i_{\ell-1}+j_{\ell-1}}v_0$ disconnects the graph, these subgraphs can be reflected across the chord edges, when the graph is regarded as an outerplanar embedding. After the reflection of all subgraphs, rescale the graph to obtain an embedding as declared in Definition 3.9. Figure 3.11 (b) shows the graph after this procedure. □

This result gives rise to the following strengthening of the notion *plane circuit*.

Definition 3.40. *A circuit C in an outerplanar graph $G = (V, E)$ is an* outerplane circuit *if there is an outerplanar embedding $G' = (V', E')$ of G in virtue of the bijection $\varphi : V \to V'$ such that $\bigcup\{\varphi(u)\varphi(v) \mid uv \in C\}$ is the boundary of some face of this embedding.*

Coming back to cycle bases, an early result concerning the MSFCB on outerplanar graphs is by Peleg and Tendler.

Theorem 3.41 ([101]). *The minimum strictly fundamental cycle basis of an unweighted outerplanar is computable in linear time.* □

The algorithm provided by the authors is fairly simple. They compute the dual graph of an outerplanar embedding and perform a breath first search rooted at v^∞ on it, where $v^\infty = \varphi_F(f^\infty)$ with φ_F defined as in Definition 3.11 for the dual graph. Also the size of a minimum SFCB is linearly bounded by the number of nodes.

Theorem 3.42 (Theorem 4.5.2 in [108]). *An unweighted outerplanar graph with n vertices has a strictly fundamental cycle basis with size $\mathcal{O}(n)$.* □

Since we deal only with 2-connected graphs, an obvious lower bound for the size of an SFCB on outerplanar graphs is $\Omega(n)$. Hence, Theorem 3.42 can be extended to

Theorem 3.43. *The size of a minimum strictly fundamental cycle basis of a 2-connected unweighted outerplanar graph with n vertices is $\Theta(n)$.* □

In the next subsection, we describe the notion of minor monotonicity which has been required to obtain this result.

3.6.3 Minor Monotonicity

An important concept which has been used to prove Theorem 3.42 is the *minor monotonicity* of the minimum size for an SFCB on unweighted outerplanar graphs. In detail, the statement of Theorem 3.42 had firstly been shown only for maximal outerplanar graphs. Afterwards, it was pointed out that the size of a minimum strictly fundamental cycle basis in an unweighted outerplanar graph decreases if an edge is removed. This property is what we call minor monotonicity. Our initial hope was to derive a priori upper bounds for minimum SFCBs on further graph classes by exploiting minor monotonicity.

In this subsection, we study the behavior of the minimum weight of an SFCB for weighted outerplanar graphs. In this process, it is figured out that we must retire to a fairly restricted subclass of weighted outerplanar graphs when we wish to preserve minor monotonicity on them. In addition to this property, it is possible to compute a minimum SFCB of a graph in this class in polynomial time.

We shall now define the term minor monotonicity in a more formal way.

Definition 3.44 (minor monotone). *Consider a graph $G = (V, E)$ and a permutation $\pi : \{1, 2, \ldots, m\} \to E$ of the edges, define*

$$G_i^\pi := \begin{cases} G, & i = m \\ G_{i+1}^\pi \setminus \pi(i+1), & \text{otherwise, i.e. } 0 \leq i < m, \end{cases} \tag{3.31}$$

and denote Φ_i^π the minimum weight of an SFCB of G_i^π. The minimum SFCB weight is minor monotone *on a class $\mathcal{G}$ of graphs if the sequence $F_G = (\Phi_1^\pi, \Phi_2^\pi, \ldots, \Phi_m^\pi)$ increases monotonically for each graph $G \in \mathcal{G}$ and each permutation $\pi \in S_m$.*

Minor monotonicity can also be considered for general cycle bases or other classes than SFCBs. We will present examples of graph classes on which even the minimum SFCB weight is not minor monotone. Thus, this result directly carries over to the superclasses of SFCBs. This is not immediately clear for classes which do not include SFCBs. Two of these classes are treated in this thesis, namely 2-bases and the different types of robust cycle bases. For the robust types we remark that the cycle basis in Example 3.46 is also strictly robust, see Chapter 4. Hence, the minimum size of a strictly robust cycle basis is not minor monotone as well. However, this is different in the case of 2-bases, cf. Definition 3.13.

Lemma 3.45. *The minimum weight of a 2-basis is always minor monotone.*

Proof. Let $B = \{C_1, \ldots, C_\nu\}$ be a 2-basis of a graph G and e an edge which shall be removed. If e is contained in only one element of B, simple removing this element from B leads to a cycle basis of $G \setminus e$ which is trivially a 2-basis with a smaller size than B. If e occurs in two circuits, say, $e \in C_1 \cap C_2$, set $B' = B \setminus \{C_1, C_2\} \cup \{C_1 + C_2\}$. The basis B' is a 2-basis of $G \setminus e$ and its size is $\Phi(B') \leq \Phi(B) - 2w(e)$. Note that the cycle $C_1 + C_2$ is possibly not a circuit. Anyway, due to Theorem 3.17, each minimum 2-basis consists only of circuits, thus, a minimum 2-basis of $G \setminus e$ could even be smaller than B'.

If the graph is directed and the arc a is deleted define the new basis $B' = B \setminus \{C_1, C_2\} \cup \{\operatorname{sign}(C_1(a)) \cdot C_1 - \operatorname{sign}(C_2(a)) \cdot C_2\}$, where $\operatorname{sign}(x)$ for a real number x is defined as usual, i.e.

$$\operatorname{sign}(x) = \begin{cases} 1, & \text{if } x > 0 \\ -1, & \text{if } x < 0 \\ 0, & \text{otherwise.} \end{cases}$$

□

The technique described in the proof above can also be interpreted as a kind of *edge-swap*, similar to the one provided in [6] to reduce the weight of a given SFCB. For a fundamental spanning tree T of a graph G, replacing a tree edge e with another edge in its fundamental cut $S_T(e)$ can result in an SFCB of smaller weight. In contrast, the edge e is additionally removed from G in our case.

As already mentioned above, the minimum SFCB weight is minor monotone on the class of unweighted outerplanar graphs. But unfortunately, it is not minor monotone even on the simple class of weighted outerplanar graphs. Yet, it can be shown that minor monotonicity can be maintained if the class is restricted to a greater extend. Altogether, three properties are necessary: the graph must be outerplanar, it must be internal face free, and its weight function must be metric. Note that all three properties are hereditary under subgraphs, i.e. they are preserved after removals of edges or vertices. Below, we provide three examples of graphs on which the minimum SFCB weight is not minor monotone because in each case, exactly one of the three properties is missing. Thus, all three are indeed necessary.

Example 3.46.
- *outerplanar*
- *internal face free*
- ***not** metric*

The weighted outerplanar and internal face free graph on the left hand side in Figure 3.12 contains six circuits with lengths $7, 7, 8, 11, 12$ and 13. The three shortest circuits are linearly independent, thus, they constitute a unique minimum cycle basis with weight 22. This basis is also strictly fundamental. The graph on the right hand side arises after deleting the unique edge which is contained in each of the three shortest circuits. The new graph contains three circuits with lengths $11, 12$ and 13. Hence, the minimum cycle basis has a length of $11 + 12 = 23$. $\Diamond$

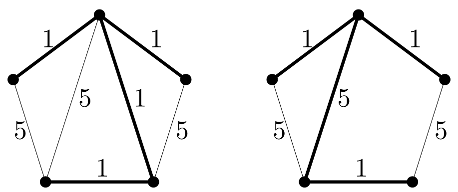


Figure 3.12: The minimum SFCB weight is not minor monotone on weighted outerplanar graphs which are additionally internal face free.

Example 3.47.
- *internal face free*
- *metric*
- ***not** outerplanar*

Consider again the left graph in Figure 3.12. Subdividing each weight 5 edge with four vertices results in an internal face free graph that is not outerplanar. Using unique weights makes the weight function metric. For the same reason as in Example 3.46, the minimum SFCB weight is not minor monotone on this graph. $\Diamond$

Example 3.48.
- *metric*
- *outerplanar*
- ***not** internal face free*

Consider the outerplanar graph depicted in Figure 3.13 (a) and initially assume it to be unweighted. A minimum SFCB is induced by the tree T which is drawn with bold edges, including the dashed one. Figure 3.13 (b) shows the dual graph and a BFS tree which is the dual tree of T, thus, T generates indeed a minimum SFCB according to Theorem 3.41 respectively to [101].

The weight function remains metric if the weight of the dashed edge of G is reduced to $\delta \in (0, 1)$. Denote this new graph by G_R. Furthermore, the weight of each circuit which

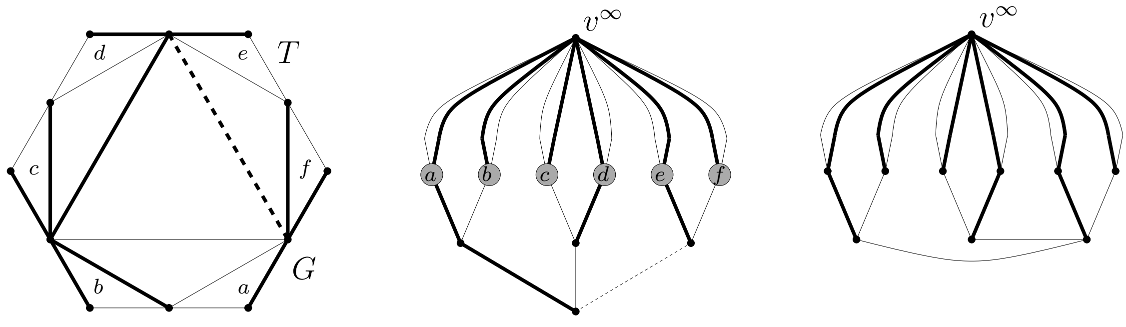


Figure 3.13: The minimum SFCB weight is not minor monotone on outerplanar graphs with internal faces, even if the weight function is metric.

contains this edge decreases by the same value. For an arbitrary minimum SFCB, each edge of G is contained in at most five basic circuits, because there is only one vertex v in the dual graph with $\text{dist}(v^\infty, v) = 3$. The length of a path in a BFS tree from v to any other vertex is thus at most 5. Hence, reducing the edge weight of the dashed edge causes the largest decrement of the basis. Therefore, the basis indicated by T is also minimum for the graph G_R. The weight of this basis on G_R is $30 + 5\delta$.

Now, remove the dashed edge to obtain the graph G'_R. The graph G'_R has unique weights, hence, the minimum SFCB can be computed by performing a BFS on the dual graph, see Figure 3.13 (c). The corresponding SFCB on G'_R has a weight of 32. Setting $\delta = \frac{1}{5}$, we get $\Phi(G_R) = 31 < 32 = \Phi(G'_R)$. ◊

A common problem in Examples 3.46 and 3.47 is that non-outerplane circuits are too short. In Example 3.48 the minimum cycle basis—the one consisting of the interior faces—is not strictly fundamental. Now, we bring the three properties together and consider outerplanar internal face free graphs with a metric weight function and achieve the statement of Theorem 3.50. In order to do this, we cite the *Exchange Theorem* from the Survey.

Theorem 3.49 (Exchange Theorem, Theorem 3.9 in [64]). *If B is a directed or undirected cycle basis of a graph G, $C \in B$ and $C = C_1 + C_2$, then either $B \setminus \{C\} \cup \{C_1\}$ or $B \setminus \{C\} \cup \{C_2\}$ is also a cycle basis of G.* □

Theorem 3.50. *The minimum SFCB weight is minor monotone on the class of internal face free outerplanar graphs with metric weights. Additionally, a minimum strictly fundamental cycle basis on this class can be computed in linear time.*

Proof. For an outerplanar and internal face free graph G with a metric weight function, consider an outerplanar embedding of this graph. Due to the metric weight function, the Hamiltonian circuit is a longest outerplane circuit. Thus, according to Theorem 3.16, the basis B consisting of the interior faces is minimum among all 2-bases. The absence of internal faces ensures that this basis is also strictly fundamental, since each face contains a Hamiltonian edge which can be seen as its private edge.

Actually, this basis is also minimum among all cycle bases. To see this, assume that there was a non-outerplane circuit $C \in B$ with $\ell > 3$ edges, regarded as a subgraph of G. The node set of C is denoted by $V(C) = \{v_0, v_1, \ldots, v_{\ell-1}\}$, while the edge set is $E(C) = \{v_k v_{(k+1) \bmod \ell} \mid 0 \leq k < \ell\}$. The circuit C is not outerplane, thus, there is a chord edge $v_i v_j \in E(G) \setminus E(C)$ with $\{v_i, v_j\} \subset V(C)$. Additionally, define the circuits $C_1 = \{v_j v_i, v_i v_{i+1}, \ldots, v_{j-1} v_j\}$ and $C_2 = \{v_i v_j, v_j v_{j+1}, \ldots, v_{i-1} v_i\}$. Since in particular the chord edge $v_i v_j$ is metric, we obtain $w(C_i) \leq w(C)$ for $i \in \{1, 2\}$. See Figure 3.14 for this setting. Now reconsider the circuits as vectors and observe $C = C_1 + C_2$. Due to the

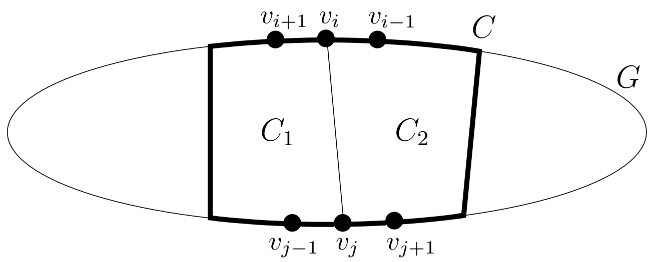


Figure 3.14: Setting of the proof for Theorem 3.50. The non-outerplane circuit is drawn with bold edges.

Exchange Theorem 3.49 either $B_1 = B \setminus \{C\} \cup \{C_1\}$ or $B_2 = B \setminus \{C\} \cup \{C_2\}$ is a cycle basis of G with a weight not greater than $\Phi(B)$. Note that the circuits C_1 and C_2 are not necessarily outerplane circuits. Anyway, the described procedure can be iterated until one obtains a strictly fundamental 2-basis B_P which consists exclusively of plane circuits with $\Phi(B_P) \leq \Phi(B)$.

With Lemma 3.45 it follows that the minimum SFCB weight is minor monotone on the class we dealt with.

It remains to show that the quoted running time can be achieved. This, in turn, is the statement of Theorem 3.16, since the basis is a 2-basis. □

The second paragraph of the proof of Theorem 3.50 does also constitute the proof of the following theorem.

Theorem 3.51. *A minimum 2-basis of an outerplanar graph G with a metric weight function is also minimum among all cycle bases of G.* □

Below the proof of Lemma 3.45 we described an edge-swap method: after the removal of a tree edge, we constructed out of a given SFCB a smaller one by replacing the deleted tree edge by another edge from the induced fundamental cut. However, despite the result of Theorem 3.50, this construction does in general only work if the basis is a 2-basis and hence actually minimum. Example 3.52 below illustrates this observation.

Example 3.52.
Consider the graph in Figure 3.15. Obviously, it is outerplanar and internal face free. Its weight function is metric if the edges without weights are assumed to have weight 1, including the edge e.

Figure 3.15: A graph with an SFCB with weight 42 (left). The graph after deleting the edge e (right). A smallest SFCB that can be constructed out of the given SFCB by an edge-swap has weight 44.

The fundamental spanning tree on the left hand side induces an SFCB with weight 42. After the removal of the edge e we are not able to construct a smaller SFCB by performing the edge-swap method. $\Diamond$

As already mentioned, the class treated in this subsection is a rather restricted one. Nevertheless, we should note that it is, to the best of our knowledge, the first non-trivial class of *weighted* graphs for which the MSFCB is solvable in linear time. However, this result mainly has its roots in the fact that the MSFCB coincides with the MINIMUM 2-BASIS Problem on this class. Moreover, this actually holds also for *all* other minimum cycle basis problems defined in this thesis. The next section is dedicated to present another class of weighted graphs for which the MSFCB is solvable in linear time. To yield this result, we do not benefit from any coincidences with other minimum cycle basis problems.

3.7 Cycle Root Graphs

The graphs given in Subsection 3.6.3 on minor monotonicity of the minimum SFCB weight had a very simple structure. If such a graph is outerplanar embedded, each interior face is adjacent to the unbounded one, i.e. to a specific face. Similarly, in a *cycle root graph*, the cycles share a specific vertex. Cycle root graphs in which parallel edges and loops are permitted, can be shown to contain the duals of outerplanar multigraphs. In spite of this, we will to define them separately.

Definition 3.53 (cycle root, cycle root graph). *A vertex r in a graph G is a* cycle root *if r is contained in every cycle of G. A graph is called* cycle root graph *or* CR-graph *if it has a cycle root.*

A cycle root graph may have more than one cycle root. For the rest of this section let G be a CR-graph and r one of its cycle roots.

Proposition 3.54. *A CR-graph is planar.*

Proof. The property of having a cycle root is hereditary under subgraphs. Thus, the planarity of CR-graphs follows simply from the fact that neither K_5 nor $K_{3,3}$ is a CR-graph. □

Lemma 3.55. *The dual of a CR-graph G is outerplanar. Possibly, the dual graph contains some parallel edges.*

Proof. With Proposition 3.54 G is planar, hence, it can be embedded into the plane. As usual, we construct the dual graph G^* by drawing one dual vertex into the interior of each face of the embedded graph G. Every face of G is bounded by a circuit and every circuit contains the cycle root. Thus, each dual vertex lies in the boundary of the face which corresponds to the cycle root. Parallel edges are generated for example if the primal graph contains vertices of degree 2. □

The aim of this subsection is to derive a class of graphs in which the MSFCB is solvable in polynomial time. As a preparation, observe that for a spanning tree T of a CR-graph G with cycle root r and an edge $uv \in E \setminus T$ we have $\mathrm{dist}_T(u,v) = \mathrm{dist}_T(u,r) + \mathrm{dist}_T(v,r)$. The reason is that for an arbitrary edge $e = uv$ in a CR-graph, each u-v-path which is not constituted by e itself has to pass the cycle root r. If G has more than one cycle root, the path passes through all of them. After summation and addition of the chord weights, it follows immediately that

$$\Phi(T) = \sum_{e=uv \in E \setminus T} \mathrm{dist}_T(u,r) + \mathrm{dist}_T(v,r) + w(e). \tag{3.32}$$

This holds for any spanning tree on a cycle root graph. Since the summation term on the right hand side of Equation (3.32) appears somewhat unhandy, we define $w_T^r(e) := \mathrm{dist}_T(u,r) + \mathrm{dist}_T(v,r) + w(e)$ for a single edge $e = uv \in E \setminus T$ and a cycle root r. The next step is to show that a fundamental spanning tree T induces a minimum SFCB if it is a shortest path tree rooted at a cycle root. In order to do this, we customize the proof of Lemma 4.5 in [101] to weighted graphs. The rough idea of the proof is substantially the same, and we borrow several notations, as well.

Theorem 3.56. *In a CR-graph $G = (V,E)$ with a positive weight function $w : E \to \mathbb{R}_+$ and a cycle root r, a shortest path tree rooted at r induces a minimum strictly fundamental cycle basis of G.*

Proof. Denote by T^* a shortest path tree rooted at r and let T be an arbitrary spanning tree. For such a tree T, let $g_r(T)$ be the number of vertices which have the smallest possible distance to r in T, i.e. $g_r(T) = |\{v \in V \mid \mathrm{dist}_T(v,r) = \mathrm{dist}_{T^*}(v,r)\}|$. Clearly, $g_r(T^*) = n$. Now, our aim is to show that $\Phi(T) \geq \Phi(T^*)$ holds for any spanning tree T. This is done by a kind of reverse induction on $g_r(T)$. More precisely, we point out in the base case that all shortest path trees rooted at r induce a strictly fundamental cycle basis of the same weight. This is possible since an arbitrary shortest path tree T can be transformed to a fixed shortest path tree T^* by performing suitable edge swaps which do not change the weight of the induced SFCB. In the second part of the proof it is shown that all other trees provide an SFCB with a weight not smaller than $\Phi(T^*)$. In detail, we show in the inductive step that $g_r(T^*) > g_r(T)$ implies $\Phi(T^*) \leq \Phi(T)$.

Base case ($g_r(T) = n$).
In a tree T with $g_r(T) = n$, each vertex has the minimum distance to r, hence, T is a shortest path tree rooted at r. Consider at first an edge $e = uv \in E \setminus (T \cup T^*)$. Then $w^r_T(e) = w^r_{T^*}(e)$ does hold since $\mathrm{dist}_T(u, r)$ and $\mathrm{dist}_T(v, r)$ as well as $\mathrm{dist}_{T^*}(u, r)$ and $\mathrm{dist}_{T^*}(v, r)$ are minimum.

Now let $e = uv \in T \setminus T^*$ with, say, $\mathrm{dist}_T(u, r) < \mathrm{dist}_T(v, r)$. In T^*, the r-v-path passes another vertex u' and another edge $e' = u'v$, of course with $\mathrm{dist}_T(r, u')$ being minimum. Also, we have $\mathrm{dist}_T(r, u') + w(e') = \mathrm{dist}_T(r, v)$.

With this edge $e' = u'v \in T^*$, it holds that $\mathrm{dist}_{T'}(v, r) = \mathrm{dist}_T(v, r)$ for $T' = T \setminus e + e'$, since all distances to r are minimum. In particular, T' is also a shortest path tree, so that $\mathrm{dist}_{T'}(v, r) = \mathrm{dist}_T(v, r)$, which implies $w^r_{T'}(e) = w^r_T(e')$. In this way, T^* can be rebuilt from T without changing any vertex distance to r and hence, obtaining $\Phi(T) = \Phi(T^*)$.

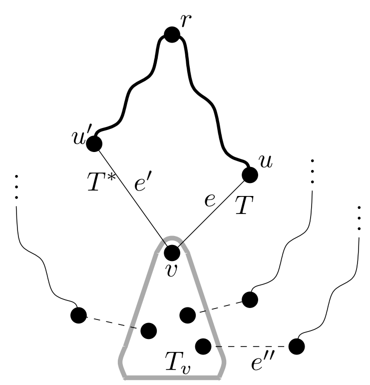


Figure 3.16: Setting of the proof of Theorem 3.56. The fat paths are in both trees T and T^*, while e' is only contained in T^* and e only in T.

Inductive step ($g_r(T) < n$).
Assume now that $\Phi(T') \geq \Phi(T^*)$ does hold for all trees T' with $g_r(T') > g_r(T)$. Since $g_r(T) < n$, there is a vertex v with $\mathrm{dist}_T(v, r) > \mathrm{dist}_{T^*}(v, r)$. Fix such a vertex v with the minimum value $\mathrm{dist}_{T^*}(v, r)$, then the r-v-paths in T and T^* must differ in their last edges. Otherwise, with uv being the common last edge, it would hold that $\mathrm{dist}_T(u, r) > \mathrm{dist}_{T^*}(u, r)$, and v was not chosen minimum. Thus, there exist edges $e' = u'v \in T^*$ and $e = uv \in T$ with $e' \neq e$ and $\mathrm{dist}_T(u', r) = \mathrm{dist}_{T^*}(u', r)$ as well as $\mathrm{dist}_T(u, r) = \mathrm{dist}_{T^*}(u, r)$; clearly, $w(e') < w(e)$. See Figure 3.16 for the setting so far.

Now, consider the tree $T' = T \setminus e + e'$ and denote the subtree of T' rooted at v by T_v, i.e. the tree $T_v = \{e = xy \in T' \mid v \in \text{Path}_{T'}(x, r)\}$. Observe that $\text{dist}_{T'}(v, r) = \text{dist}_{T^*}(v, r)$ and thus, $g_r(T') \geq g_r(T) + 1$. From $w(e') < w(e)$ we conclude that $w^r_{T'}(e) < w^r_T(e')$. Additionally, there can be several edges e'' with one end node in $V(G[T_v])$ and with $w^r_{T'}(e'') < w^r_T(e'')$. Such edges e'' are drawn dashed in Figure 3.16. Note that there are no edges with both end nodes in $V(G[T_v])$, because G is a CR-graph. Altogether, the distances decrease and we yield $\Phi(T) \geq \Phi(T')$ and, by the inductive hypothesis, $\Phi(T) \geq \Phi(T^*)$. $\square$

Since a shortest path tree can be computed in polynomial time, e.g. by performing Dijkstra's algorithm, we conclude

Theorem 3.57. *The* MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS *Problem is solvable in polynomial time on the class of weighted cycle root graphs.* $\square$

An additional consequence of Theorem 3.56 arises when we put the duality of CR-graphs and outerplanar graphs together with the duality of the MSFCB and the GOCST, defined in Subsection 3.4.3.

Theorem 3.58. *The* GENERAL OPTIMUM COMMUNICATION SPANNING TREE *Problem is solvable in polynomial time on weighted outerplanar graphs.* $\square$

3.8 Conclusions

In this chapter, we investigated the MINIMUM STRICTLY FUNDAMENTAL CYCLE BASIS Problem on planar graphs and on several subclasses of planar graphs. The problem emerged as $\mathcal{NP}$-complete on planar graphs, and thus, we turned our focus to the subclasses. By taking a closer look on weighted outerplanar graphs, we were able to construct a class of weighted graphs on which the MSFCB is solvable in linear time—the first one to the best of our knowledge. We were also able to find a second class.

The chapter also motivates the investigation of some open questions. The complexity status of the MSFCB is still open for arbitrary weighted outerplanar graphs. At the end of the introduction we mentioned upper bounds for the size of a minimum SFCB. Has any graph a strictly fundamental cycle basis with a size in $\mathcal{O}(m \log n)$? What is about the "edge-swap" heuristic by Amaldi et al.? Some experiments let us conjecture that it is an approximation algorithm on general graphs and an exact algorithm on weighted outerplanar graphs.

Chapter 4

Classification of Robust Cycle Bases

The following chapter is concerned with robust cycle bases. In Section 4.1 we motivate why it is valuable to investigate this type of cycle bases. Even though not much research had been done on this topic, we start with listing what is known up to now. A practical and a theoretical application is described in Section 4.2. Cyclically robust cycle bases and three further types are defined in Section 4.3 in a stringent way. This section closes with a short list of known results on these classes. In Section 4.4, examples of different types of robust cycle bases are elaborately provided. We carry on the investigation on the relationship of robust cycle bases with fundamental cycle bases in Section 4.5.

Contribution. After the emergence of quasi-robust and strictly quasi-robust cycle bases in [99], several straightforward inclusions of the different types can be deduced. We draw the outcoming map of robust cycle bases and fill it with examples of suitable graphs and cycle bases. This completes the classification of robust cycle bases.

One focus in [67] is the relationship of robust and fundamental cycle bases. We continue this research by providing more examples of cycle bases which are even minimum in almost all cases. We are able to eliminate one of two question marks in a map given there, where the authors conjectured the existence of examples. However, this does not show that robustness and fundamentality are totally unrelated concepts, as assumed in [67], but it intensifies this impression, at least.

4.1 Introduction

When dealing with cycle bases of graphs, one discovers that the cycle space contains more than circuits. In the directed case, the entry of a vector in the cycle space can take any value, because of the possibility to scale a vector arbitrarily. Beside this, also in the undirected case cycles with nodes of degree four or more and not connected cycles may appear. The distinction between circuits and cycles which are no circuits is the most important matter in the area of robust cycle bases.

Assume that we would like to reconstruct a circuit from the circuits of a cycle basis. This can be done by subsequently adding up the basic circuits one by one. When doing so, it is possible that each partial sum is also a circuit. Clearly, this depends on the order in which the basic circuits are added. If there is such an order for each circuit, we speak about *cyclically robust cycle bases*. Strengthening and weakening of this concept lead to four classes of robust cycle bases, overall.

When trying to take a look back into the history of robust cycle bases, one realizes that not much research has been done on the field of robust cycle bases. Indeed, a query at the online data base *Zentralblatt MATH* ([131]) produced only three results on "robust cycle bases" in the title. They were introduced as "a new type of cycle basis for graphs" in 2002 in an article from Kainen ([63]). Already in 1976 it had been conjectured by Dixon and Goodman ([37]) that every strictly fundamental cycle basis is cyclically robust, though the authors did not use this terminology. In their paper, they proposed an algorithm for the problem of finding a longest cycle in a graph. The conjecture was disproved in 1979 by Sysło in [119], where the author additionally presented several now well-known characterizations for strictly fundamental cycle bases. The approach of comparing robust cycle bases with fundamental cycle bases was picked up in 2009 by Klemm and Stadler ([67]). Also in 2009, the idea not to use only basic circuits from the support of a given circuit led to the concept of quasi-robust cycle bases, see [99].

Unfortunately, not much more results concerning robust cycle bases than these ones mentioned in the last paragraph were published. Several classes of graphs which admit robust cycle bases of various types are listed in the tabular at the end of Section 4.3. Moreover, Proposition 2 and Conjecture 1 in [63] proposed two more graph classes to be cyclically robust, namely the complete bipartite graphs and the cartesian product graph $G \times T$ under the assumption that G has a cyclically robust cycle basis and T is a tree. Anyway, both proposals were disproved in [98].

4.2 Applications

The concept of robust cycle bases admits also some applications. We describe two of them slightly more precisely. Additionally to these applications, "quasi-robustness is of interest in its own right in the context of sampling algorithms on the cycle space: quasi-robustness is necessary and sufficient for ergodicity of the Markov chains considered in [66]", see [99]. For further applications we refer e.g. to [63] and [67].

Retrieval in Chemical Databases. In large chemical databases, molecular structures are usually represented by graphs. When searching for such structures, one is faced with the problem of deciding whether graphs are isomorphic, thus, a problem whose complexity is still unresolved. One way to considerably reduce the search space is to exclude a large number of graphs by investigating their cyclic structure. Amongst others, the "set of all elementary cycles", i.e., in our language, the set of all *circuits*, plays an important role for this process.

To construct the set of all circuits, cycle space algorithms linearly combine circuits of a given cycle basis, and save the resulting cycle if it is a circuit. To reach actually all circuits of a graph in this way, it is necessary and sufficient that the basis is cyclically robust.

For more details on this application see [13] and the references therein, especially [39]. In this survey, the authors describe which other sets of cycles besides the set of all circuits are used to examine the cyclic structure of graphs representing molecular structures.

Category Theory. Kainen ([63]) gave an application of strictly robust cycle bases in category theory. To describe it, we need at first several definitions. A *category* is a class of objects like sets or topological spaces together with a class of morphisms for every pair of objects. Morphisms can be for example functions or homeomorphisms. A category is *small* if its objects are sets. Furthermore, it is possible to compose morphisms. This composition has to obey the axioms of associativity and identity, which we do not specify at this place. A map between two categories is referred to as *functor*. Objects and morphisms of a category are usually represented as *diagrams* where vertices stand for the objects and the arcs between them for the morphisms. Such a diagram can essentially be seen as a digraph. A *groupoid* is a category in which each morphism is an isomorphism. When considering groupoids, we thus may restrict to graphs instead of digraphs.

Let CAT be the category with all small categories as objects and the functors between them as the morphisms. In this category, a morphism between two functors is called a *natural transformation*. A *natural equivalence* is a natural transformation which is an isomorphism.

Now consider a subcategory of CAT which is a groupoid, a diagram of it, and a circuit in this diagram. Take two vertices v and w from this circuit and look at the two paths between v and w along the circuit. Since we consider a groupoid, the path compositions induce functors. If there is a natural equivalence between these functors, then we call the circuit *commutative up to natural equivalence*. A whole diagram is *commutative up to natural equivalence* if each of its circuit is.

Still, an efficient way to test a diagram for commutativity up to natural equivalence is missing. Therefore, we need Lemma 1 from [63] which states: "Suppose that two cycles in the underlying graph of a diagram intersect in a nontrivial path. If both cycles are commutative up to natural equivalence, then so is their sum.". In our context, cycles have to be regarded as circuits. It follows that a diagram commutes up to natural equivalence if each circuit in a strictly robust cycle basis commutes up to natural equivalence. Clearly, this would be much more efficient than checking each circuit. But unfortunately, no method is known for constructing a strictly robust cycle basis.

4.3 Classes of Robust Cycle Bases

In this section, we define four different types of robust cycle bases. In order to do this, we essentially follow the completions in [99]. Similarly as there, we need at first the concept of (strictly) well-arranged sequences of circuits. Afterwards, we deduce several simple inclusions and present a map of the relationship between the different classes of robust cycle bases.

Definition 4.1 ((strictly) well-arranged sequence). *A sequence $S = (C_1, \ldots, C_k)$ of circuits in an undirected graph is called* well-arranged *if for all $j = 1, \ldots, k$ the sum $\sum_{i=1}^{j} C_i$ is also a circuit. A well-arranged sequence of circuits is* strictly well-arranged *if for all $j = 2, \ldots, k$ the intersection $C_j \cap \sum_{i=1}^{j-1} C_i$ is a single path.*

The path in Definition 4.1 contains at least one edge. Otherwise, the sum $C_j + \sum_{i=1}^{j-1}$ was not a circuit and thus, the sequence was not even well-arranged, at all. It is clear that every strictly well-arranged sequence is also well-arranged. Furthermore, it is known that there are well-arranged sequences that are not strictly well-arranged. The authors of [67] provide such an example in which the sum of two circuits is again a circuit, but they intersect in three paths. Note that it is not forbidden that a circuit appears more than once in a (strictly) well-arranged sequence.

With this in mind we are now able to define the four different variants of robust cycle bases which were developed in [99].

Definition 4.2 (cyclically/strictly robust and (strictly) quasi-robust cycle bases). *A cycle basis B of a graph G is* (strictly) quasi-robust *if for each circuit C in G there is a (strictly) well-arranged sequence $S_C = (C_1, \ldots, C_{k-1}, C_k)$ such that $C = \sum_{i=1}^{k} C_i$ and $C_i \in B$ for $i = 1, \ldots, k$. A strictly quasi-robust cycle basis is* strictly robust *if the circuits in the strictly well-arranged sequence are pairwise disjoint. Similarly, a quasi-robust cycle basis is* cyclically robust *if the according well-arranged sequence does not contain a circuit twice. If we do not want to specify the particular type of robustness, we simply speak about a* robust cycle basis.

It can be concluded that for strictly and for cyclically robust cycle bases the well-arranged sequence of a circuit C must not contain basic circuits which are not in the support of C. Also, directly from these definitions, we can immediately derive the following facts:

- every strictly quasi-robust cycle basis is quasi-robust,

- every strictly robust cycle basis is strictly quasi-robust,
- every strictly robust cycle basis is cyclically robust, and
- every cyclically robust cycle basis is quasi-robust.

These inclusions hold since in each case, we require additional properties for the more specific class. The inclusions give rise to the diagram in Figure 4.1.

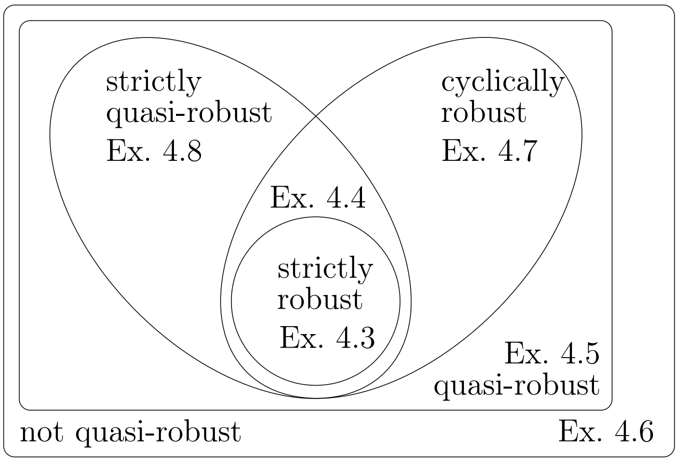


Figure 4.1: Map of robust cycle bases.

Not much is known about which graph classes can have which type of robust cycle bases. Further it is unknown whether each graph admits a robust cycle basis of any of the four types. The table below summarizes the related results. To the best of our knowledge, these are the only known ones.

Graph class	Robustness	Reference
planar graphs	strictly robust	[38]
complete graphs	strictly robust	[63]
complete bipartite graphs $K_{m,n}$ with $m \leq 4$ and $n \leq 5$	strictly robust	[99]
general complete bipartite graphs	quasi-robust	[99]
wheels	cyclically robust	[67]

4.4 Examples of Robust Cycle Bases

In this section, we show that the inclusions derived in the last section are valid only in the given direction. Thus, no two of the classes are equivalent. To point this out, we give an example of a graph and a cycle basis for each region in the map in Figure 4.1 and thus show that it is not empty.

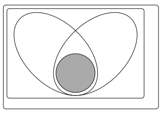
Except in Example 4.4, all cycle bases are strictly fundamental. According to the Weight Lemma (Lemma 3.6), we can choose a weighting function such that this cycle basis is the unique minimum cycle basis on this graph. However, the given cycle basis in Example 4.4 is also the unique minimum one. The existence of a graph with a minimum cycle basis in each region of the map indicates that each class—actually even each non-empty difference of two classes—of robust cycle bases admits its own minimization problem.

Remember that we do not know an efficient algorithm for the computation or for the recognition of any type of robust cycle bases on general graphs. Thus, to prove a cycle basis of a graph G as (strictly) quasi-robust, we have to indicate a (strictly) well-arranged sequence of basic circuits for every circuit in G. Analogous sequences have to be found for (strictly) robust cycle bases. In the latter case, a basic circuit is allowed to occur at most one time in each of these sequences.

On the other hand, a cycle basis B of a graph G is not quasi-robust if there exists a circuit C' in G such that for each $C \in B$ the sum $C' + C$ is not a circuit. To show that the cycle basis is not strictly quasi-robust, one has to verify that the cut $C' \cap C$ does not form a path for one circuit C' in the graph and for all $C \in B$. Finally, to show that a cycle basis is not a cyclically robust or a strictly robust cycle basis it suffices to check only the circuits of the support of such a circuit C'.

We now start with the description of the examples.

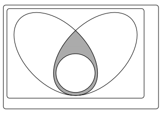
Example 4.3.



- *strictly robust*

The first example is the simple graph C_3 that consists of exactly one circuit of length 3. Clearly, its unique cycle basis is strictly robust—and strictly fundamental and minimum, as well. $\diamondsuit$

Example 4.4.



- *cyclically robust*
- *not strictly robust*
- *strictly quasi-robust*

Our second example is the complete bipartite graph $K_{3,3}$, see Figure 4.2 (a). The cycle basis $B = \{C_1, C_2, C_3, C_4\}$ is highlighted in Figure 4.2 (b). It is not strictly fundamental, thus, we suggest the indicated weighting to make the cycle basis minimum. All other circuits have a greater weight. The weights of all circuits are denoted below the graph in Figures 4.2 (b), (c) and (d). We show that B is cyclically robust and strictly quasi-robust, but not strictly robust. For $2 \leq k \leq \nu$ we denote $C_{i_1,\ldots,i_k} := \sum_{j=1}^{k} C_{i_j}$.

Cyclically Robust. The $K_{3,3}$ is cubic, hence it contains no cycle with vertices of degree 4 or more. Furthermore, it has only six vertices, but it is triangle free. Thus, there is no

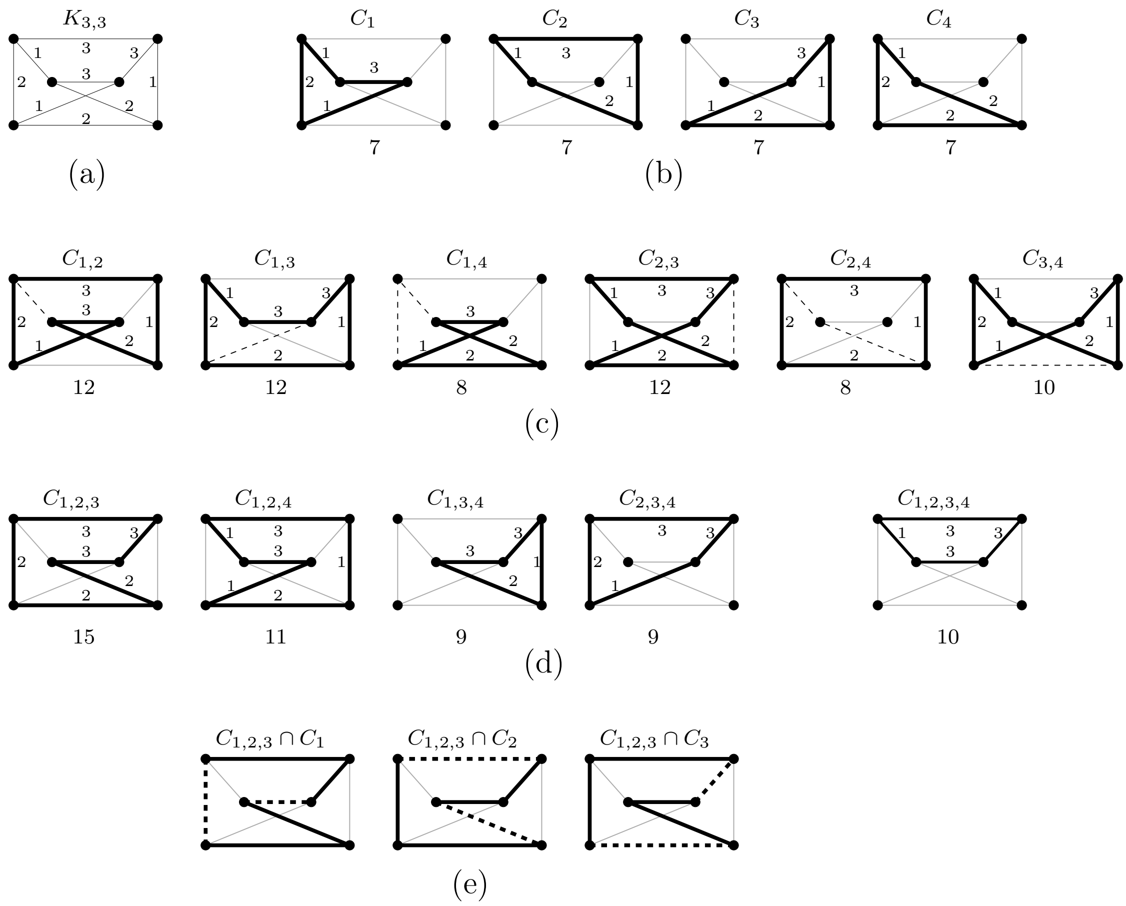


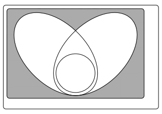
Figure 4.2: The $K_{3,3}$ with weights on the edges (a). The four basic circuits and their weights below (b). All other circuits and their weights (c) and (d). The intersections (dashed edges) of $C_{1,2,3}$ with the circuits of its support (e).

cycle consisting of two triangles. This means that every cycle is a circuit and therefore, each cycle basis of $K_{3,3}$ is cyclically robust.

Not Strictly Robust. The given basis is not strictly robust, since there is no strictly well-arranged sequence for $C_{1,2,3}$, in which every basic circuit occurs only once. Observe this by looking at Figure 4.2 (e). It is indicated that $C_{1,2,3}$ has an intersection consisting of two path (dashed edges) with each circuit from its support.

Strictly Quasi-Robust. For the circuits which have exactly two basic circuits in their supports, these two circuits intersect in a single path, illustrated by the dashed edges in Figure 4.2 (c). For the circuits depicted in Figure 4.2 (d) we provide the sequences $S_{C_{1,2,3}} = (C_1, C_3, C_4, C_2, C_4)$, $S_{C_{1,2,4}} = (C_1, C_4, C_2)$, $S_{C_{1,3,4}} = (C_1, C_3, C_4)$, $S_{C_{2,3,4}} = (C_2, C_4, C_3)$, and $S_{C_{1,2,3,4}} = (C_1, C_3, C_4, C_2)$, which are all strictly well-arranged. Hence, the cycle basis is strictly quasi-robust. $\Diamond$

Example 4.5.



- *quasi-robust*
- *not cyclically robust*
- *not strictly quasi-robust*

This example is borrowed from [63]. We consider the complete bipartite graph $K_{5,5}$, the strictly fundamental cycle basis B induced by the spanning tree T shown in Figure 4.3 (a), and the circuit C aside in Figure 4.3 (b). The sixteen basic circuits themselves are also depicted in Figure 4.4 as black edges. Assigning weights according to the Weight Lemma 3.6, B becomes the unique minimum cycle basis.

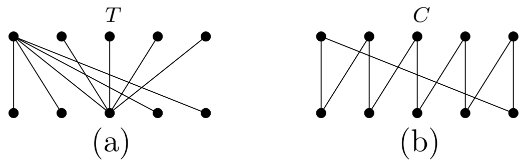


Figure 4.3: Spanning tree T of $K_{5,5}$ (a). The circuit C considered in the text (b).

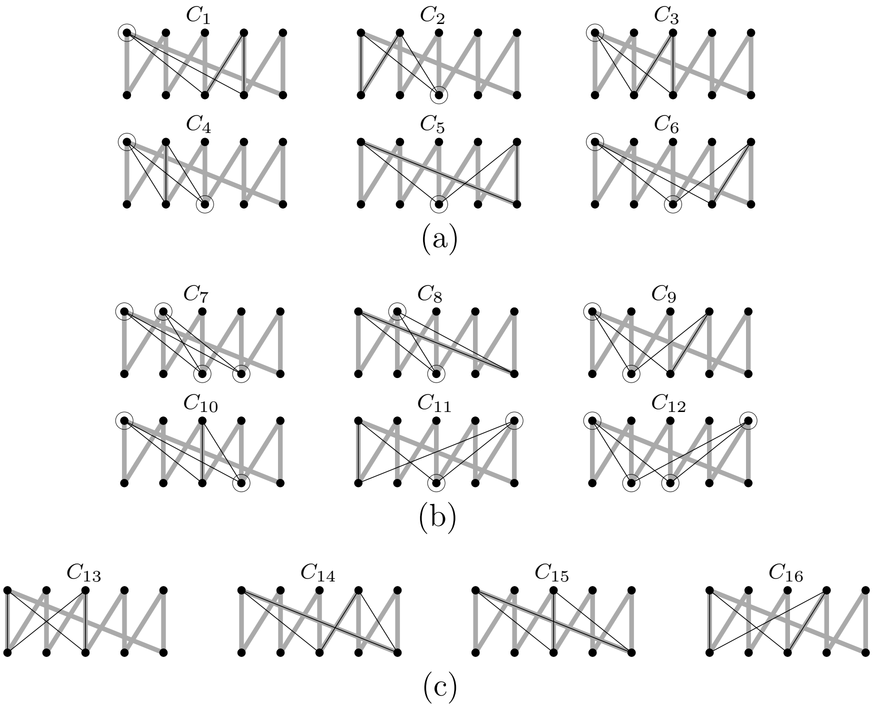


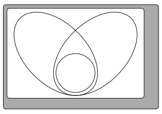
Figure 4.4: The sixteen basic circuits of B (black edges) and the circuit C (grey edges). For the sake of clearness we dropped the edges which are neither in the basic circuit nor in C.

Quasi-Robust. The described basis had been shown to be quasi-robust in [99] in an elaborate manner.

Not Cyclically Robust. The circuit C can be written as $C = \sum_{i=1}^{6} C_i$ and all the sums $C + C_i$ for $i = 1, \ldots, 6$ are cycles with node degrees greater than 2 (marked by a circle), see Figure 4.4 (a). Hence, this cycle basis is not cyclically robust.

Not Strictly Quasi-Robust. Looking at the remaining basic circuits we observe that also C_7 to C_{12} yield cycles with node degrees of 4, Figure 4.4 (b), again marked by a circle. The intersection of C_{13} to C_{16} with C is not a single path in each case, as can be seen in Figure 4.4 (c). In addition, $C + C_{13}$ and $C + C_{14}$ are disconnected. All in all, the cycle basis is not strictly quasi-robust. $\diamondsuit$

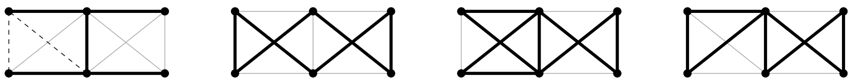
Example 4.6.



- *not quasi-robust*

The example of a cycle basis which is not even quasi-robust presented here had been inspired by a talk of Ostermeier ([98]).[1]

The cycle basis is strictly fundamental and it is induced by the bold drawn tree in Figure 4.5 (a).

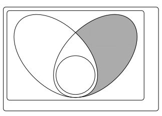


(a) (b) (c)

Figure 4.5: Graph with an inducing fundamental spanning tree (a), a circuit C (b), and sums of C with two basic circuits generated by the dashed chords (c).

Not Quasi-Robust. Due to symmetry we have to consider only the basic circuits induced by the dashed edges. In both cases, they add up with C to a cycle that is not a circuit, see Figure 4.5 (c). $\diamondsuit$

Example 4.7.



- *cyclically robust*
- *not strictly quasi-robust*

The next example is a cycle basis on Wagner's graph V_8 which is cyclically robust, but not strictly quasi-robust. The strictly fundamental basis is indicated by the spanning tree which is highlighted on the left hand side in Figure 4.6 (a). The basic circuits are denoted at the chords. We use the notation from Example 4.4, i.e. $C_{i_1,\ldots,i_k} := \sum_{j=1}^{k} C_{i_j}$.

[1] A similar example already appeared in [119].

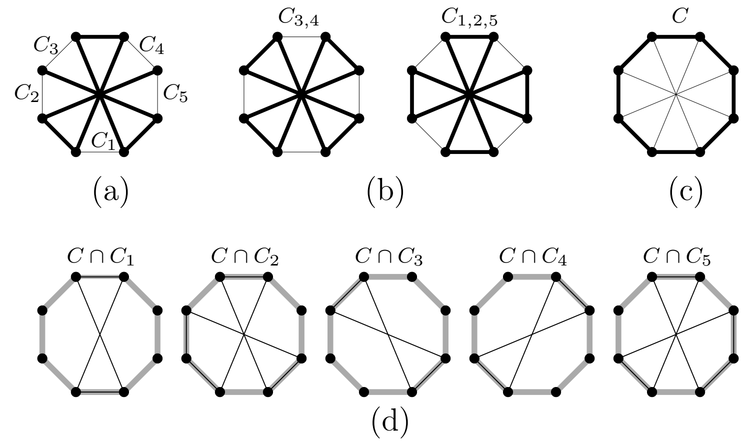
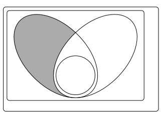


Figure 4.6: Wagner's graph V_8 with a fundamental spanning tree (a). The only two non-circuits in V_8 (b). The circuit C (c). The intersections of C (grey) with the five basic circuits do not form a single path (d), edges which are not in a basic circuit or in C are dropped.

Cyclically Robust. Wagner's graph V_8 is cubic which implies that every cycle is 2-regular. The only critical cycles in V_8 are thus the two non-circuit pictured in Figure 4.6 (b). We provide the well-arranged sequences $S_{C_{3,4}+C_1} = S_{C_{1,3,4}} = (C_1, C_3, C_4)$, $S_{C_{3,4}+C_2} = S_{C_{2,3,4}} = (C_2, C_3, C_4)$, and $S_{C_{3,4}+C_5} = S_{C_{3,4,5}} = (C_4, C_5, C_3)$ for the circuits which arise by adding a remaining basic circuit to $C_{3,4}$. For the cycle $C_{1,2,5}$ we give the sequences $S_{C_{1,2,5}+C_3} = S_{C_{1,2,3,5}} = (C_1, C_2, C_3, C_5)$ and $S_{C_{1,2,5}+C_4} = S_{C_{1,2,4,5}} = (C_1, C_2, C_4, C_5)$. In each of these sequences, every basic circuit appears at most once. This shows that the basis B is cyclically robust.

Not Strictly Quasi-Robust. To see that the basis is not strictly quasi-robust, consider the circuit C in Figure 4.6 (c). Its intersection with each basic circuit does not form a single path. This is illustrated in Figure 4.6 (d). $\Diamond$

Example 4.8.



- *strictly quasi-robust*
- *not cyclically robust*

The last example provides a graph with a cycle basis $B = \{C_1, \ldots, C_6\}$ which is strictly quasi-robust but not cyclically robust. As in Example 4.4 denote $C_{i_1,\ldots,i_k} := \sum_{j=1}^{k} C_{i_j}$ for $2 \le k \le \nu$.

Strictly Quasi-Robust. Since $\nu = 6$ we have to investigate $2^6 - 6 - 1 = 57$ cycles; the six basic circuits and the zero vector are not interesting. The 22 cycles listed below are not circuits.

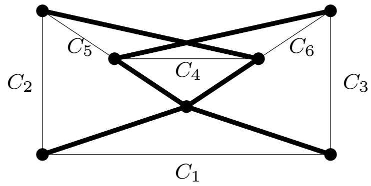


Figure 4.7: Graph with a fundamental spanning tree which induces a cycle basis that is strictly quasi-robust but not cyclically robust.

$$
\begin{array}{llllllll}
C_{1,4}, & C_{2,3}, & C_{1,5,6}, & C_{4,5,6}, & C_{1,4,5,6}, & C_{2,3,5,6}, & C_{1,2,3,4,5}, & C_{1,2,4,5,6}, \\
C_{1,5}, & C_{1,4,5}, & C_{2,4,6}, & C_{1,2,4,6}, & C_{2,3,4,5}, & C_{2,4,5,6}, & C_{1,2,3,4,6}, & C_{1,3,4,5,6}, \\
C_{1,6}, & C_{1,4,6}, & C_{3,4,5}, & C_{1,3,4,5}, & C_{2,3,4,6}, & C_{3,4,5,6} & &
\end{array}
$$

For the remaining eleven circuits $C_{i,j}$ with $|\operatorname{supp}(C_{i,j})| = 2$ we may ignore the order of the basic circuits. The intersection of the two basic circuits is a path in each case, and thus, the sequences are strictly well-arranged. For the 24 circuits with at least three elements in their supports, we provide the following strictly well-arranged sequences.

$$
\begin{array}{llll}
(C_1, C_2, C_3), & (C_1, C_3, C_6), & (C_4, C_6, C_3), & (C_1, C_2, C_5, C_6), \\
(C_1, C_2, C_4), & (C_3, C_4, C_2), & (C_5, C_6, C_3), & (C_4, C_6, C_3, C_1), \\
(C_1, C_2, C_5), & (C_3, C_5, C_2), & (C_1, C_2, C_3, C_4), & (C_5, C_6, C_3, C_1), \\
(C_1, C_2, C_6), & (C_3, C_6, C_2), & (C_1, C_2, C_3, C_5), & (C_1, C_2, C_3, C_5, C_6), \\
(C_1, C_3, C_4), & (C_4, C_5, C_2), & (C_1, C_2, C_3, C_6), & (C_1, C_2, C_3, C_6, C_5, C_4, C_1), \\
(C_1, C_3, C_5), & (C_2, C_5, C_6), & (C_4, C_5, C_2, C_1), & (C_1, C_2, C_3, C_5, C_6, C_4).
\end{array}
$$

Not Cyclically Robust. Figure 4.8 illustrates that the treated cycle basis is not cyclically robust. More precisely, look at the circuit $C_{2,3,4,5,6}$. For $i = 2, \ldots, 6$, the cycles $C_{2,3,4,5,6} + C_i$ have nodes with degree greater than 2, marked by circles in Figure 4.8. Hence, this circuit does not admit a well-arranged sequence in which the circuits are pairwise disjoint.

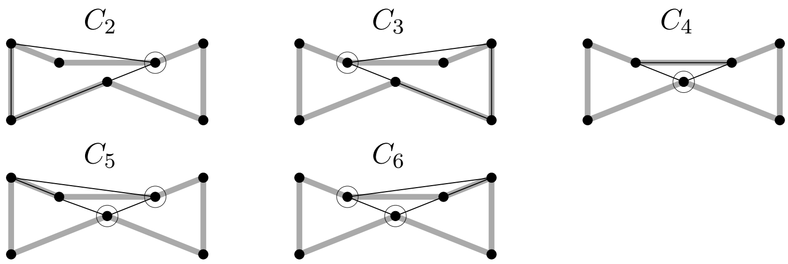


Figure 4.8: The circuit $C_{2,3,4,5,6}$ (grey) and the five basic circuits of its support (black).

$\Diamond$

4.5 Relationship with Fundamental Bases

One approach for a better understanding of strictly robust and cyclically robust cycle bases had been presented in [67]. Therein, the authors investigated the relationship between strictly robust, cyclically robust, and non-robust cycle bases on one hand, and strictly fundamental, weakly fundamental, and non-fundamental cycle bases on the other hand. Their motivation was the detailed exploration of strictly and weakly fundamental cycle bases which had been done in the years before. They concluded that robustness and fundamentality of cycle bases "are essentially unrelated concepts".

In more detail, they considered the combination ($\texttt{robustness}, \texttt{fundamentality}$), where $\texttt{robustness} \in \{$"strictly robust", "robust", "non-robust"$\}$ and $\texttt{fundamentality} \in \{$"strictly fundamental", "weakly fundamental", "non-fundamental"$\}$. This immediately led to nine possibilities, and an example of a graph with an according cycle basis was presented in seven of these cases.

In this section, we follow up this line of research and provide for eight cases a graph with an appropriate cycle basis which is additionally minimum. Anyway, we do not reinvent the wheel and borrow in some cases graphs and cycle bases which already served as examples in other contexts. For the ninth case, we are able to retire to a strictly quasi-robust cycle basis instead of a strictly robust one. However, this basis is not the minimum basis of the presented graph.

We start with three examples of strictly fundamental bases. Two of them are taken from [67], the third one is taken from Section 4.4. Due to the Weight Lemma, all bases can be made minimum.

Example 4.9.

- *strictly robust*
- *strictly fundamental*

This example is directly taken from [67]. To be more accurate, we deal with the complete graph K_n and the cycle basis B_n which is induced by the complete bipartite graph $K_{1,n-1}$ as *fundamental spanning tree*. It is *strictly robust* as shown in [63]. With a weighting assigned according to the Weight Lemma it is also the *unique minimum* cycle basis.

We decided to present this example here because it constitutes a whole class of graphs and cycle bases with the required properties. On the other hand, also the triangle graph in Example 4.3 could have served as an example at this place. $\Diamond$

Example 4.10.

- *not strictly robust*
- *cyclically robust*
- *strictly fundamental*

Wagner's graph V_8 and the cycle basis which had already been presented in Example 4.7 provide the necessary properties for this example. We remark that this example eliminates one of the two question marks in [67] where the authors conjectured the existence of such an example. $\diamond$

Example 4.11.

- *non-robust*
- *strictly fundamental*

Again, we borrow the example given in [67] which is called there "Ostrowski's basis". It is simply the K_5 with a path consisting of four edges as *fundamental spanning tree*. This spanning tree induces a basis consisting of three triangles, two quadrangles, and one pentagon. To verify that the basis is *non-robust*, take a look at the circuit C which is the sum of the three triangles and the two quadrangles. The sum of C with each of these basic circuits constitutes a non-circuit.

Similarly to Example 4.10, we could have borrowed the graph with a non-robust cycle basis from Example 4.6. Anyway, we used Ostrowski's basis at this place because there is an easy way to construct an infinite class of graphs and cycle bases with the required properties. More precisely, we speak about the family of complete graphs with an odd number of vertices. For such a graph $G_k = (V_k, E_k)$ with $V_k = \{v_0, v_1, \ldots, v_{2k}\}$ we choose the path $(v_0, v_1, \ldots, v_{2k})$ as inducing spanning tree for the strictly fundamental cycle basis. As a certificate for the non-robustness, we provide the circuit $C_k = \bigcup_{i=0}^{2k}\{v_i v_{i+2}\} = \sum_{i=0}^{2k}\{v_i v_{i+1}, v_{i+1}v_{i+2}, v_{i+2}v_i\}$, where the indices are taken modulo $2k+1$. Adding one basic circuit $C_k^i = \{v_i v_{i+1}, v_{i+1}v_{i+2}, v_{i+2}v_i\}$ to C_k results in a cycle C_k' with $\deg_{C_k'}(v_{i+1}) = 4$. In Figure 4.9, the graph G_3 is given as an example.

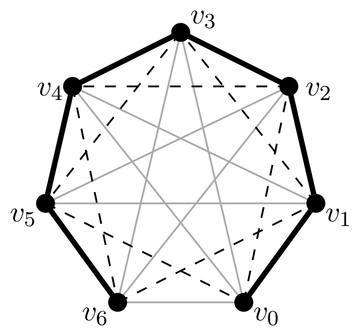


Figure 4.9: The graph G_3, the inducing spanning tree (fat edges), and the circuit C_k (dashed edges).

$\diamond$

We continue with three examples which are weakly fundamental but not strictly fundamental. One example is taken from [78]. For the other two, we destroy the strictly fundamentality of the according examples above by gluing suitable graphs together. In doing so, we have to keep in mind that we want the bases to stay minimum.

Example 4.12.

- *strictly robust*
- *not strictly fundamental*
- *weakly fundamental*

The additional demand for a minimum cycle basis prevents us from simply copying the according example in [67]. Instead, we copy Example 11.2 from [78], which deals with the sunflower graph SF(3), depicted in Figure 4.10. In [78], it served as an example for a 2-basis which is not strictly fundamental.

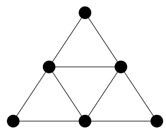


Figure 4.10: The sunflower graph SF(3).

The planar cycle basis B consisting of the only four triangles is the unique minimum cycle basis. Regarding SF(3) as an outerplanar graph, consider the circuit formed by the three chord edges. Each of its edges is contained in another circuit of B, thus, it is *not strictly fundamental*. But since the basis is a 2-basis, it is *weakly fundamental* according to Lemma 3.15 and *strictly robust* due to [38]. Actually, everything above does also hold for all outerplanar graphs with a metric weight function, because the minimum 2-basis of a graph from this class is also minimum among all cycle bases, see Theorem 3.51. $\Diamond$

Example 4.13.

- *not strictly robust*
- *cyclically robust*
- *not strictly fundamental*
- *weakly fundamental*

The idea in this example is to adapt Wagner's graph and its cycle basis presented in Examples 4.7 resp. 4.10 such that it is not strictly fundamental anymore. To do this, we append a further path (v_1, v_2, v_3) at the two adjacent vertices v_1 and v_3 at the right hand side of the graph, see Figure 4.11.

The weights of the graph are assigned according to the Modified Weight Lemma (3.7). The second statement of this lemma does hold for v_1v_3, i.e. $w(v_1v_3) < \mathrm{dist}_T(v_1, v_3)$. For the new edges set $w(v_1v_2) = w(v_2v_3) = 1$. To yield the cycle basis, inherit the basic circuits from the original example and append the circuit $C_6 = (v_1v_2, v_2v_3, v_3v_1)$. Remark that the weights of the old edges were chosen according to the Modified Weight Lemma (Lemma 3.7) and that C_6 is the shortest circuit which contains the new vertex v_2. Hence, the obtained cycle basis is minimum.

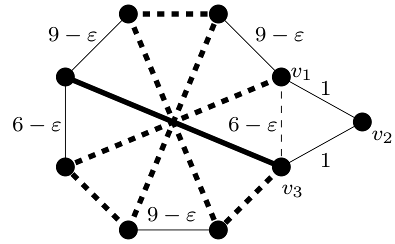


Figure 4.11: The modified Wagner's graph with a partial spanning tree (fat edges) and a circuit without private edge (dashed).

The basis is *not strictly robust* for the same reasons as in Example 4.7. On the other hand, assume that a circuit C in this graph contains the vertex v_2. A well-arranged sequence for C can be achieved by concatenating C_6 with the well-arranged sequence of $C+C_6$, hence, the basis is *cyclically robust*. Finally, the cycle basis is *not strictly fundamental*, since the dashed basic circuit does not have a private edge. But it is *weakly fundamental* because Inequality (3.1) holds for each permutation π with $C_{\pi(6)} = C_6$. ♢

Example 4.14.

- *non-robust*
- *not strictly fundamental*
- *weakly fundamental*

Similarly to the example above, we destroy the strictly fundamentality of Ostrowski's basis of the K_5. We also could have used the Modified Weight Lemma and could have constructed a graph by simply appending a path of length 2 as in Example 4.13. Anyway, we decided to provide a larger example in favor of an integer weight function.

Remember that the basis of this graph was induced by a path of four edges as fundamental spanning tree. There is one edge between the end nodes of the path, denote it e_P. Now take three copies of K_5 and assemble them in a way such that the three copies of e_P constitute a triangle, add a vertex and connect it to the three corners of the triangle. See Figure 4.12 for the construction. The edge weights in the three copies of K_5 are assigned according to the Weight Lemma, the three new edges get the weight 2. Again, the fat edges get weight 1.

To get a cycle basis for the merged graph, combine the cycle bases of the three copies and append the three new triangles with weight 8, i.e. the triangles constituted by two new edges and one copy of e_P. The $\nu = 21$ shortest circuits have weight 8, hence, the combined cycle basis is minimum. It is *not robust* because Ostrowski's basis is not. It is not *strictly fundamental* since the circuits induced by e_P in each K_5 do not have private edges, as well as the three new triangles. In the end, it is *weakly fundamental*. Permute the basis such that the three new triangles appear at first, followed by the three circuits induced by the copies of e_P. ♢

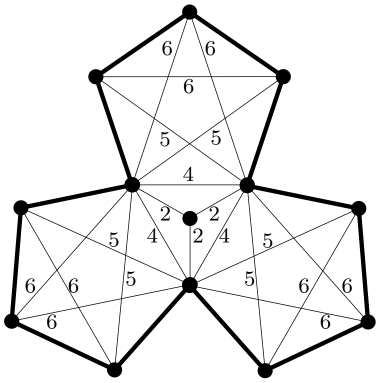


Figure 4.12: Three merged copies of K_5 with Ostrowski's bases.

The last three examples present non-fundamental cycle bases. Two of them are again borrowed from [78].

Example 4.15.

- *strictly quasi-robust*
- *non-fundamental*

Unfortunately, we were not able to give an example of a minimum non-fundamental cycle basis which is strictly robust. But we provide a strictly quasi-robust one, at least. Therefore, look at the graph depicted in Figure 4.13 and the indicated cycle basis.

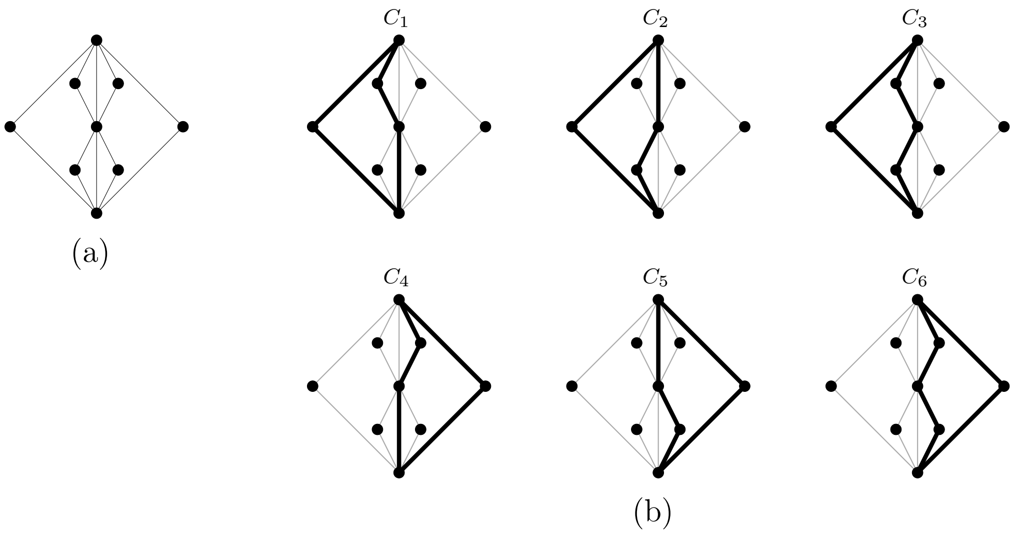


Figure 4.13: A graph (a) and a non-fundamental cycle basis which is strictly quasi-robust, but not strictly robust (b).

The basis is *non-fundamental* since each edge is contained in at least two basic circuits. To see that it is *strictly quasi-robust*, we take a look at $2^6 - 6 - 1 = 57$ cycles, analogous to Example 4.8. Among these cycles, there are 38 which do not constitute circuits. For the other 19 circuits, we provide the strictly well-arranged sequences below.

(C_1, C_3),	(C_1, C_3, C_4),	(C_1, C_3, C_4, C_6),	$(C_6, C_5, C_2, C_3, C_1)$,
(C_2, C_3),	(C_6, C_4, C_1),	(C_2, C_3, C_5, C_6),	$(C_1, C_3, C_4, C_6, C_5)$,
(C_4, C_6),	(C_2, C_3, C_5),	$\boldsymbol{(C_6, C_4, C_1, C_3, C_2, C_5, C_6)}$,	$(C_2, C_3, C_5, C_6, C_4)$,
(C_5, C_6),	(C_6, C_5, C_2),	$(C_1, C_3, C_4, C_6, C_2)$,	$(C_6, C_4, C_1, C_3, C_2, C_5)$,
(C_3, C_2, C_1),	(C_6, C_5, C_4),	$\boldsymbol{(C_3, C_1, C_4, C_6, C_5, C_2, C_3)}$	

For the circuits which belong to the bold written sequences, there are no strictly well-arranged sequences in which all circuits are pairwise disjoint. Thus, the cycle basis is strictly quasi-robust, but not strictly robust. $\Diamond$

Example 4.16.

- *not strictly robust*
- *cyclically robust*
- *non-fundamental*

The cycle basis in this example is borrowed from [78] where it serves as an example of a minimum cycle basis which is not integral[2]. It is a basis of the generalized Petersen graph $P_{11,4}$, see Figure 4.14.

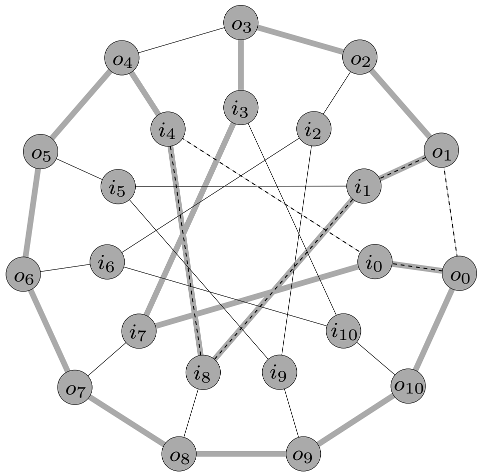


Figure 4.14: Generalized Petersen graph $P_{11,4}$ with the basic circuit C_1 (dashed) and the circuit $C_{1,4,12} = C_1 + C_4 + C_{12}$ (grey).

The discussed basis B contains the circuits $C_{j+1} = (o_j i_j,\ i_j i_{j+4},\ i_{j+4} i_{j+8},\ i_{j+8} i_{j+1},\ i_{j+1} o_{j+1},\ o_{j+1} o_j)$ for $j = 0, \ldots, 10$ where the indices are taken modulo 11, and the circuit $C_{12} = \{o_0 o_1, \ldots,\ o_9 o_{10},\ o_{10} o_0\}$. In the figure above we emphasized the circuit C_1 with dashed edges. With the weights $w(o_j o_{j+1}) = 4$, $w(i_j i_{j+4}) = 5$, and $w(o_j i_j) = 12$, again for $j = 0, \ldots, 10$ and again modulo 11, this basis becomes the unique minimum one, see [78].

Each edge $i_j i_{j+4}$ is contained in three basic circuits, all other edges in exactly two basic circuits. This shows the *non-fundamentality* of the basis. To see that it is *not strictly*

[2] For the definition of integral cycle bases we refer to Section 5.3.

robust, consider for example the circuit $C_{1,4,12} = C_1 + C_4 + C_{12}$ whose cuts with C_1, C_4, and C_{12} do not form a single path in each case.

It remains to show that the basis is cyclically robust. This was done by a small program implemented in C++ using LEDA ([89]). The program tested for each of the 2^{12} linear combinations if it constitutes a circuit C, and if so, if there is a circuit $C_j \in \operatorname{supp}(C)$ such that $C + C_j$ is a circuit. This applied to each circuit and thus, the cycle basis is *cyclically robust*. $\Diamond$

Example 4.17.

- *non-robust*
- *non-fundamental*

To construct cycle bases of a biconnected graph which are neither robust nor fundamental, the authors in [67] suggest the following operation. Given a 2-connected graph G' with a non-robust cycle basis B' and a 2-connected graph G'' with a non-fundamental cycle basis B'', construct a graph G by identifying two arbitrary edges of G' and G''. The basis $B = B' \cup B''$ is a basis of G. However, even if B' and B'' are the minimum cycle bases of G' and G'', respectively, it is not guaranteed that B is a minimum cycle basis of G. In contrast to this construction, we propose Champetier's graph with its minimum cycle basis as a representative for a minimum non-robust and non-fundamental cycle basis.

Also this graph and the cycle basis are taken from [78]. In his Example 11.7, Liebchen considered Champetier's graph whose unique minimum cycle basis is integral but neither weakly fundamental nor totally unimodular. In Champetier's original paper [23], it served as a counter-example of a conjecture expressed in [40]: "If G is null-homotopic (i.e., if every cycle of G is the modulo 2 edge sum of triangles), there is an edge e of G such that $G \setminus e$ is still null-homotopic.". Champetier's graph is visualized in Figure 4.15.

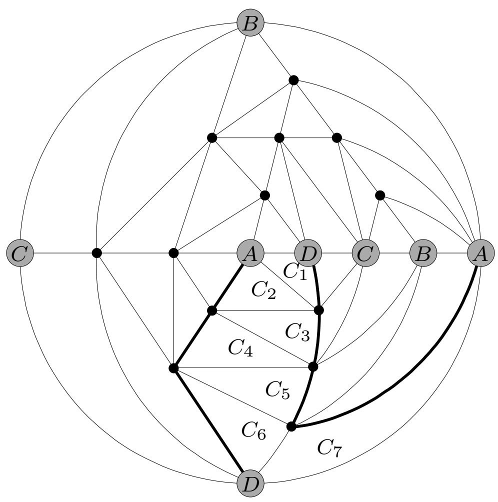


Figure 4.15: Champetier's graph and a certificate for the non-robustness of the minimum cycle basis.

Champetier's graph arises from the embedding by identifying the vertices A, B, C and D with their copies. The cycle basis we deal with is formed by the 36 triangles in the embedded version. This basis is minimum since there is neither a further triangle which is not the boundary of a face in the embedding in Figure 4.15 nor a path of length 3 between two copies of one of the vertices A to D. After the vertex identifications, such a path would also compose to a triangle. Hence, the basic circuits are the only triangles.

Since each edge is contained in two triangles at least, the basis is non-fundamental. As a proof for the non-robustness, we take the same certificate as in Example 11.7 in [78], i.e. the circuit $C = \sum_{i=1}^{7} C_i$, indicated in Figure 4.15 by two paths. In fact, $C + C_i$ does not form a circuit for $i = 1, \dots, 7$. This shows that the basis is non-robust. $\diamondsuit$

The table below summarizes the results of this section. It has been inspired by the Venn diagram in [67] which also illustrates the relationship between fundamental and robust cycle bases. In the table, we contrast our results with the results listed there. New examples and improvements are emphasized in italic.

	strictly fundamental	**weakly fundamental**	**non-fundamental**	
strictly robust	K_n with $K_{1,n-1}$ as fund. sp. tree	Fig. 2 in [67]	?	[67]
	minimum basis	basis not minimum		
	as above	sunflower graph $SF(3)$	Ex. 4.15, basis only strictly quasi-robust	this thesis
	minimum basis	*minimum basis*	basis not minimum	
cyclically robust	?	Kainen's basis of K_4	non-fundamental basis of the 4-wheel	[67]
		basis not minimum	basis not minimum	
	Wagner's graph with a P_7 as fund. sp. tree	*Wagner's graph joined up with a triangle*	*Petersen graph $P_{11,4}$*	this thesis
	minimum basis	*minimum basis*	*minimum basis*	
non-robust	K_5 with P_4 as fund. sp. tree	Vogt's example	merging non-rob. basis with non-fund. basis	[67]
	basis not minimum	basis not minimum	basis not minimum	
	as above	*three merged K_5*	*Champetier's graph*	this thesis
	basis minimum with a suitable weighting	*minimum basis*	*minimum basis*	

4.6 Conclusions

This chapter was concerned with robust cycle bases. With the support of suitable examples, we were able to isolate strictly and cyclically robust cycle bases, as well as the newer concepts of quasi-robust and strictly quasi-robust cycle bases from each other. Since each of the examples provides a unique minimum cycle basis, we can view this classification of robust cycle bases as completed.

The second main result in this chapter was the continuation of the comparison between robust and fundamental types of cycle bases. We were able to further fill the Venn diagram of robust and fundamental cycle bases given in [67], where we demanded in addition that the provided cycle basis is minimum. Our results were summarized in a table which has only one missing item. We could not present a minimum cycle basis which is non-fundamental and strictly robust, but could provide an example of a cycle basis which is strictly quasi-robust, at least.

Obviously, there is still plenty of work to do in the field of robust cycle bases. At first, the gap mentioned above should be closed. For Example 4.4, the question arises whether there is a graph and a cycle basis with the three properties on the robustness considered there, which is additionally strictly fundamental. Moreover, it is still not known whether each graph provides a strictly robust cycle basis, or a cycle basis of any other robust type, at least. As a first step into this area, one could present further graph classes providing such bases. And finally, there is nothing known about the complexity of recognition and construction of robust cycle bases. It is easy to see that the problem of recognizing a cycle basis as strictly robust is in co-$\mathcal{NP}$. The author conjectures that it is even co-$\mathcal{NP}$-complete.

Chapter 5

Further Classes of Cycle Bases

This last chapter summarizes miscellaneous results on further classes of cycle bases which had not been taken into account, yet. To be more precise, in Section 5.1, we introduce p-bases as a generalization of planar bases. Section 5.2 deals with totally unimodular cycle bases. Totally unimodularity is a concept originally from the area of integer linear programming. Finally, in Section 5.3, integral cycle bases are treated. Integral cycle bases arose from the practical application of cyclic timetabling.

Contribution. Since p-bases are a new concept which is first defined in this thesis, the whole section about this topic can be seen as contribution. For an overview of the results we refer to the introduction of Section 5.1.

In contrast, totally unimodular and integral cycle bases had already been investigated in many other publications. This leads to different notions of totally unimodular cycle bases. We partially clarify the relationship between four different possible definitions for totally unimodular cycle bases. Moreover, we discover an error in an example for a totally unimodular cycle basis which is not weakly fundamental. This error is repaired by giving a new example. The new basis contains cycles which are not circuits. Such a cycle can be decomposed into circuits. A smaller cycle basis can be achieved by replacing the cycle with one of these circuits. We call this operation the Exchange Property. It is known that the Exchange Property is closed when we restrict to weakly fundamental cycle bases, i.e. the cycle can be replaced by a circuit such that the achieved cycle basis is also weakly fundamental. We show with an example that this is not true for totally unimodular cycle bases, in general. Additionally, we investigate the restriction where only circuits are allowed for a cycle basis and observe that under further restrictions weakly fundamentality of a cycle basis implies totally unimodularity.

Also for integral cycle bases, more than one definition is possible. We present three of them and show their equivalence. Furthermore, we show that there are integral cycle bases consisting of arbitrarily large cycles. This is in contrast to the imagination that basic cycles have to be kept small to ensure the integrity of the coefficients in the representation of a circuit. As a last result, we give a weaker form of the Exchange Property in terms of totally unimodular and integral cycle bases.

5.1 p-Bases

The notion of p-bases is a natural generalization of 2-bases, which were defined in Subsection 3.4.1. Recall that for a 2-basis, it had been required that each edge is in at most two basic circuits. For a p-basis, this *two* is replaced by an arbitrary natural number p. Additionally, there should also be an edge that is actually contained in p basic circuits.

For now, the new concept of p-bases is of purely theoretical interest. That is why we dispense with a historical outline or a paragraph about applications of p-bases. We start with the definition of p-bases in Subsection 5.1.1, where we also consider the difference between the directed and undirected case. In Subsection 5.1.2, it is shown that each biconnected graph has a ν-basis, where ν is the cyclomatic number. On the other hand, it can be shown that for each p, there is a graph that does not have a p-basis. This is the issue of Subsection 5.1.3.

5.1.1 Definition of p-Bases for Directed and Undirected Graphs

In this subsection, we define p-bases for undirected graphs as a generalization of 2-bases. This is further generalized to p-bases for directed graphs. We also discuss why it is reasonable to consider only p-bases which contain only circuits.

Definition 5.1 (p-basis). *A cycle basis B of an undirected 2-connected graph G is called a p-*basis *if*

1. *each edge is contained in at most p cycles of B and*
2. *there is at least one edge which is contained in exactly p cycles of B.*

For $p = 2$, we get exactly the definition of a 2-basis in [81], when we exclude the trivial cases where $\nu \leq 1$. On the other hand, the maximal p for which a graph G can have a p-basis is obviously $p = \nu(G)$.

The concept of p-bases in Definition 5.1 can be extended to directed graphs. We further generalize this notion to bases which contain cycles that are not necessarily simple.

Definition 5.2. *Let B be an arbitrary cycle basis of a directed graph and Γ the corresponding $m \times \nu$ cycle matrix. Then B is called a p-*basis *if for the row-sum norm the following* *equation holds*

$$\|\Gamma\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{\nu} |\gamma_{ij}| = p. \tag{5.1}$$

Note that a directed graph D can have a p-basis for arbitrary large p. In particular, $p > \nu(D)$ is possible, in contrast to the case of undirected graphs. Note further, that with this definition, also a non-planar graph can have a 2-basis (see Example 11.10 in [78]). For this reason, in Definition 3.1 in [64] it is additionally required that a 2-basis is an undirected cycle basis. Since we do not want to deal with p-bases for arbitrarily large p, we restrict ourselves to the undirected case respectively to cycle bases that contain only simple cycles for the rest of this section.

5.1.2 p-Bases for Large p

When we take a look at the problem of the existence of a p-basis for an arbitrary $p \in \mathbb{N}$, it becomes clear that for each p, one can construct a graph G and a cycle basis B of G such that B is a p-basis. Therefore, look at the simple example of a graph that consists of p triangles which have one edge in common.

Example 5.3.
The graph $G = (V, E)$ with $V = \{u, v, w_1, \ldots, w_p\}$ and $E = \{uv, uw_1, \ldots, uw_p, vw_1, \ldots, vw_p\}$ has the cycle basis $B = \{\{uv, vw_i, uw_i\} \mid i \in \{1, \ldots, p\}\}$, which is a p-basis. Of course, this graph is planar and hence, it has also a 2-basis. $\lozenge$

The example above can be extended to every graph. More precisely, it holds

Lemma 5.4. *For each biconnected graph $G = (V, E)$ and each edge $e \in E$ there is a $\nu(G)$-basis B such that e is contained in every element in B.*

Proof. The proof is constructive. Consider an arbitrary cycle basis $B = \{S_1, \ldots, S_\nu\}$ of G and choose a cycle that contains the edge e, say S_1. Then $B' = \{S'_1, \ldots, S'_\nu\}$ with $S'_1 = S_1$, $S'_i = \mu_i S_1 + S_i$ for $i = 2, \ldots, \nu$ and

$$\mu_i = \begin{cases} 1, & e \notin S_i, \\ 0, & e \in S_i \end{cases}$$

is such a cycle basis as desired. To see the linear independence of B', choose a cycle $S = \sum_{i=1}^{\nu} \lambda_i S_i$. With the new basic cycles, S has the representation

$$S = \left(\lambda_1 - \sum_{i=2}^{\nu} \lambda_i \mu_i\right) S'_1 + \sum_{i=2}^{\nu} \lambda_i S'_i.$$

$\square$

The basis constructed above has the drawback that it may contain some non-circuits. However, Lemma 5.4 can be strengthened.

Lemma 5.5. *For each biconnected graph $G = (V, E)$ there is a $\nu(G)$-basis that contains only circuits. The edge which is contained in each basic circuit can be chosen arbitrarily. This basis is additionally weakly fundamental.*

In order to prove Lemma 5.5 we need a proposition and a theorem. Proposition 5.6 is borrowed from [36], Theorem 5.7 from [17]. Compact proofs can be found in these cited books. In this context, if H is a graph, then an *H-path* is a u-v-path P with length$(P) \geq 1$ that shares exactly its end nodes u and v with H.

Proposition 5.6 (Proposition 3.1.3. in [36]). *A graph is 2-connected if and only if it can be constructed from a circuit by successively adding H-paths to graphs H already constructed.* □

The decomposition that arises from the construction in this proposition is sometimes referred to as *(proper) ear decomposition.* It is originated from [124]. Secondly, we need the well-known Menger Theorem ([90]).

Theorem 5.7 (Menger Theorem, Theorem III.5.(i) in [17]). *Let s and t be distinct non-adjacent vertices of a graph. Then the minimal number of vertices separating s from t is equal to the maximal number of independent s-t-paths.* □

Proof of Lemma 5.5. Denote $e = uv$ the edge which shall be contained in each circuit of the basis. The graph G is 2-connected, so we can choose a circuit C_1 that contains e. On one hand, C_1 is the first circuit in B, on the other hand, it is viewed as the starting circuit of the construction suggested in Proposition 5.6. Consider now one step in this construction: for an already constructed subgraph H, an H-path P_H is added to H. This operation increases the cyclomatic number by 1, because length(H) edges and length$(H)-1$ vertices are appended to the graph. Furthermore, the H-path contains at least one new edge $f = xy$. Choose a circuit C that contains e and f and append C to B. To see the existence of such a circuit use Theorem 5.7. Therefore, add two auxiliary vertices s and t and edges su, sv, tx and ty to $H \cup P_H$. Then, there are two disjoint s-t-paths from which the desired circuit can be constructed. Just delete s and t and add e and f to the remaining parts of the paths.

Since each increment of the cyclomatic number comes along with a new circuit, B contains ν circuits. Each new circuit contains an edge, that is not contained in a circuit that is already in B. So B is linearly independent and thus a cycle basis. Due to the construction, B is weakly fundamental. □

At this point, the question whether every graph has a ν-basis of a special type is solved for each upper class of weakly fundamental cycle bases. So we may ask what happens when considering the two subclasses, namely strictly fundamental cycle bases and 2-bases.

Strictly Fundamental Cycle Bases. A strictly fundamental cycle basis is a ν-basis only if its inducing spanning tree T contains an edge whose induced cut contains all chords of T. We will see that there are graphs which do not have such a tree. As a counter-example, we choose a planar graph in order to be able to construct the dual graph.

Lemma 5.8. *A planar graph G has a spanning tree T that contains an edge whose induced cut contains all chords of T if and only if the dual graph G^* is Hamiltonian.*

Proof. It is well known that there is a one-on-one correspondence between the cuts in a connected planar graph and the circuits in its (topological) dual graph, see for example [36]. Furthermore, if T is a spanning tree in G, then the dual edges that correspond to $E \setminus T$ form a spanning tree in G^* ([3]). Consider now a Hamiltonian circuit C^* in G^* and choose one edge $e^* \in C^*$. Since C^* is Hamiltonian, $C^* \setminus e^*$ is a spanning tree in G^*. Hence, the corresponding cut in G contains all chords of T. $\square$

Exemplarily, take a look at the graph in Figure 5.1, its dual G^*, and a certificate that G^* is not Hamiltonian. With Lemma 5.8, we conclude that G does not have a strictly fundamental ν-basis.

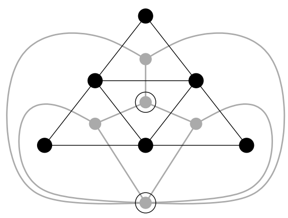


Figure 5.1: A graph G with its dual G^* (grey) and a proof that G^* is not Hamiltonian (dual vertices marked by a circle; the removal of these two vertices separates the graph into three parts, what shows that the graph is not 1-tough and hence not Hamiltonian).

2-Bases. Although this is a more or less trivial case, we decided to consider it either way, since it leads to a continuative question. Take at first a look at the already mentioned algebraic characterization of planar graphs.

Theorem 3.14 ([87])
A 2-connected graph has a planar basis if and only if it is planar. $\square$

With Definition 5.1 in mind, a 2-connected graph is planar if and only if it has a 1-basis or a 2-basis. From Theorem 3.14 we trivially conclude that a planar graph has a ν-basis only if $\nu \in \{1, 2\}$.

On the other hand, it follows immediately from Theorem 3.14 that a non-planar graph cannot have a 2-basis, so the question arises whether there are graphs which cannot have a p-basis for any other $p > 2$. This will be discussed in the next subsection.

5.1.3 p-Bases for Small p

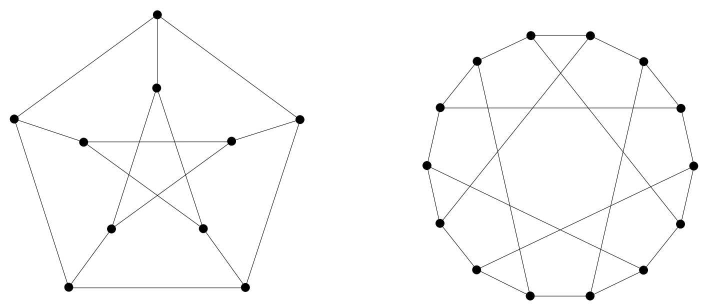
In this subsection, we consider the following problem: what is the minimal number p for which a given graph has a p-basis? Clearly, a 2-connected graph has a 1-basis only if it is a circuit. Furthermore, a 2-basis is only possible when the graph is planar. But similar to the case of non-planar graphs which cannot have 2-bases one may ask whether there are graphs without p-bases for $p > 2$.

Intuitively, a graph with this property should have a minimum cycle basis with a large size, while it has few edges. To obtain a long minimum cycle basis, we decided to consider graphs with large girths. Furthermore, we only want to deal with regular graphs to hold our computation simple. A regular graph with a specific girth that has as few edges as possible is known as a *cage*. Therefore, we give a short introduction into the topic of cages.

Cages. In the last decades, there has been a lot of interest into cages. Representatively, we refer to the dynamic survey [43] that "presents the results of over 50 years of searches for cages".

A scope of application for cages is for example coding theory. In particular, cages are used for the construction of high-rate low-density parity-check codes for magnetic recording ([86]). We proceed with the definition of cages and two examples of cages in Figure 5.2.

Definition 5.9 (Cage). *A k-regular graph with girth g and minimum number of nodes is called a* (k, g)-cage.



(a) (b)

Figure 5.2: The Petersen graph (a) and the Heawood graph (b) are the unique $(3, 5)$-cage and the unique $(3, 6)$-cage, respectively.

The lemma below ensures the existence of a cage for each reasonable pair of regularity and girth.

Lemma 5.10 (cf. [110]). *For each $k \geq 3$ and each $g \geq 3$ there is a k-regular graph with girth g.* □

This, in turn, guarantees the existence of a cage with these parameters. Note that in general, there can be more than one cage for a given pair of regularity and girth. Actually, there are three non-isomorphic $(3, 10)$-cages, see [10, 95, 127], where the latter paper also shows that there are no more $(3, 10)$-cages.

We keep at the $(3, 10)$-cage, but turn our focus back to the question whether there is a graph without a 3-basis. A $(3, 10)$-cage has 70 vertices and since it is cubic, 105 edges. Thus, the cyclomatic number is $\nu = 105 - 70 + 1 = 36$. The girth is 10 per definition . Theoretically, a minimum cycle basis of a $(3, 10)$-cage has a weight of $\nu g = 360$. Indeed, this value is attained by the minimum cycle basis of the cage constructed in [10]. But because $360/105 > 3$, there must be an edge which is contained in at least 4 basic circuits. Thus, the $(3, 10)$-cages do not provide 3-bases.

Proposition 5.11. *There are graphs which do not provide 3-bases.* □

The case $p > 3$. Taking the result of Proposition 5.11 into account, this paragraph is dedicated to show that for each $p \in \mathbb{N}$ there is a graph which does not have a p-basis. That is, Definition 5.1 does really make sense. Since there are some nice results concerning 3-regular cages, we focus our considerations to 3-regular graphs. For reasons of simplicity, we will restrict to cages with an even girth. A similar calculation can be done when considering $(3, g)$-cages with odd g.

At first, we give the following bound from [43], which has been originated in [112]. For a given girth g, let $n(g)$ be the number of vertices of a $(3, g)$-cage, $m(g) = \frac{3}{2} \cdot n(g)$ its number of edges, and $\Phi(g)$ the theoretical size of a minimum cycle basis, i.e. the product of the girth and the cyclomatic number.

Lemma 5.12 (Theorem 7 in [43]). *For every even $g \geq 4$, it holds*

$$n(g) \leq \frac{29}{12} 2^{g-2} + \frac{4}{3}. \tag{5.2}$$

□

We introduce a further variable s in Inequality (5.2) to yield an equation. For $s \geq 0$ we obtain

$$\begin{aligned}
n(g) &= \frac{29}{12} 2^{g-2} + \frac{4}{3} - s, \\
m(g) &= \frac{29}{8} 2^{g-2} - \frac{3}{2} s + 2, \text{ and} \\
\Phi(g) &= \frac{29}{24} g 2^{g-2} - \frac{1}{2} sg + \frac{5}{3} g + g.
\end{aligned}$$

Similarly to the last paragraph in the context of the $(3, 10)$-cage, we consider the quotient of the theoretical weight of a minimum cycle basis and the number of edges and obtain

$$\frac{\Phi(g)}{m(g)} = \frac{1}{3} g \frac{29 \cdot 2^g - 48s + 256}{29 \cdot 2^g - 48s + 64} \in \Theta(g).$$

Since the quotient grows linearly with the girth and, according to Lemma 5.10, there is a $(3, g)$-cage for each arbitrarily large girth, we can conclude

Theorem 5.13. *For each $p \in \mathbb{N}$, there is a graph that does not have a p-basis.* □

5.1.4 Conclusions

In this section, we introduced the concept of p-bases as a generalization of planar bases. We showed that every graph has a ν-basis which is weakly fundamental. A further restriction to strictly fundamental is not always possible. On the other hand, for each $p \in \mathbb{N}$, there is a graph which does not have a p-basis.

Naturally, the question arises whether for a given graph G and an integer p, the graph G does have a p-basis. We assume this problem to be $\mathcal{NP}$-complete. This, in turn, would lead to the question about the approximability of this problem, or to its tractability on restricted graph classes. Clearly, also many other directions of research on this area are thinkable.

5.2 Totally Unimodular Cycle Bases

In this section, we consider the class of totally unimodular (TUM) cycle bases. At a first glance, it seems to be the most artificial or unnatural class of cycle bases in this thesis. Why it is nevertheless valuably to investigate them should become clearer in Subsection 5.2.1. In the literature, different notions of totally unimodular cycle bases appeared. These are summarized and compared in Subsection 5.2.2. Finally, in Subsection 5.2.3, we detect a mistake in Chapter 11 of [78], namely an Example of a cycle basis which should be totally unimodular but not weakly fundamental. We give a new example of a cycle basis which contains non-circuits and show that under further restrictions, there is no example of such a cycle basis which consists only of circuits.

5.2.1 Introduction

In the broad area of integer linear programming, the concept of totally unimodular matrices plays an important role. An extensive introduction to the topic of totally unimodular matrices can be found in [113]. As accomplished there, if the input of a linear program is integer, the constraint matrix is totally unimodular, and the program has an optimum solution, then there is also an optimum integral solution. This, in turn, is based on the fact that for such a program, each vertex of the feasible region has integer coordinates.

Also in [113], a relaxation method for solving systems of linear inequalities is introduced. More precisely, if the appropriate matrix is totally unimodular, then the system can be solved with this method in polynomial time ([88, 113]).

Section 5.3 will deal with integral cycle bases. As it will be mentioned there, the complexity status of the Minimum Integral Cycle Basis Problem is still unknown. As an approach to attack this problem, subclasses of integral cycle bases had been investigated. Natural subclasses from a combinatorial point of view are strictly and weakly fundamental cycle bases. Both minimum cycle basis problems restricted to these classes are algorithmically intractable, see [32] and [52]. As we will see in the next section, the determinant of an integral cycle basis is 1. Hence, from the perspective of the cycle matrix, totally unimodular bases form a natural subclass of integral cycle bases. Note that the complexity status of the Minimum Totally Unimodular Cycle Basis Problem is also still unknown. But since totally unimodular matrices are well studied, it is convenient to investigate the class of totally unimodular cycle bases as a subclass of integral cycle bases.

5.2.2 Basic Definitions and Properties on TUM Cycle Bases

In the last years, several notions of totally unimodular cycle bases emerged in the literature. In this subsection, we provide an overview of the different definitions and compare them. Meanwhile, we are able to partially solve Open Problem 3 in [64], which asks for the relation between these definitions. Clearly, when considering totally unimodular cycle bases, one has to know what a totally unimodular matrix is, at first.

Definition 5.14 (totally unimodular matrix, cf. [113]). *A matrix is* totally unimodular *if each square submatrix has a determinant in* $\{-1, 0, +1\}$.

The proposition below follows immediately.

Proposition 5.15. *A matrix* Γ *is not totally unimodular if there is a submatrix of* Γ *which is not totally unimodular.* □

In Subsection 5.2.3, we will profit from the following characterization of totally unimodular matrices.

Lemma 5.16. *A matrix is totally unimodular if and only if each subset of its columns can be split into two parts such that the sum of the columns in one part minus the sum of the columns in the other part is a vector with entries in* $\{-1, 0, +1\}$. □

For the proof of this lemma and further characterizations for totally unimodular matrices we refer again to [113]. Also from there, we list up several operations which preserve totally unimodularity. The second item immediately implies that Lemma 5.16 can also be formulated with rows. Note that we skip to list up further operations which are not needed in this section.

Lemma 5.17. *A totally unimodular matrix remains totally unimodular after*

1. *permuting rows or columns,*
2. *transposition,*
3. *multiplying a row or a column by* -1*,*
4. *adding a row or a column with only one non-zero element, or*
5. *repeating a row or a column.* □

Now let us turn our focus back to cycle bases. We list up four different possible definitions for totally unimodular cycle bases that can be found in the literature. Because there cannot be different definitions for the same term, we use an own notation.

Definition 5.18 (Γ-TUM, Definition 10.5 in [78]). *A cycle basis of a directed graph is* Γ-TUM *if its cycle matrix* Γ *is totally unimodular.*

Definition 5.19 (G-TUM, Definition 3.1 in [64]). *A cycle basis* $B = \{C_1, \ldots, C_\nu\}$ *of a directed graph is* G-TUM *if each cycle* C' *of* $G(D)$ *has an orientation* C *that can be written as a linear combination with coefficients in* $\{-1, 0, +1\}$ *of circuits in* B*, i.e.*

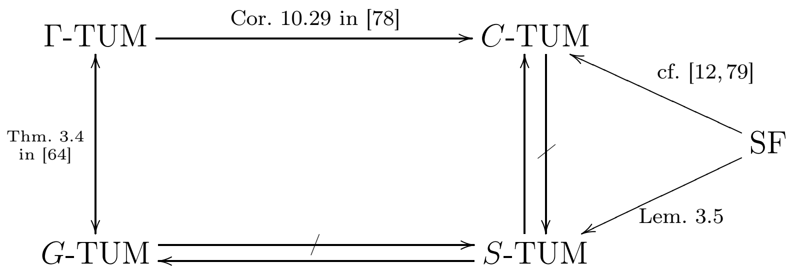
$$\exists \lambda_i \in \{-1, 0, +1\} : C = \sum_{i=1}^{\nu} \lambda_i C_i.$$

Also in [64], two other definitions had been suggested with the motivation that they seem to be more natural than Definition 5.19.

Definition 5.20 (C-TUM, cf. Open Problem 3 in [64]). *A cycle basis* B *of a directed graph is* C-TUM *if each circuit* C *can be written as a linear combination with coefficients in* $\{-1, 0, +1\}$ *of circuits in* B*.*

Definition 5.21 (S-TUM, cf. Open Problem 3 in [64]). *A cycle basis* B *of a directed graph is* S-TUM *if each simple cycle* S *can be written as a linear combination with coefficients in* $\{-1, 0, +1\}$ *of circuits in* B*.*

These definitions are summarized in the diagram below. Here, also known and new relations between the definitions are illustrated. Additionally, the definition of strictly fundamental cycle bases is integrated in this diagram. An arrow in the diagram points from a subclass to a superclass of cycle bases.



The two unlabelled implications are discussed below. Moreover, we give some notes on several redirections.

S-TUM $\Longrightarrow$ C-TUM. This does obviously hold since each circuit is also a simple cycle.

S-TUM $\Longrightarrow$ G-TUM. This implication is also nearly trivial. Each simple cycle admits a $\{-1, 0, +1\}$ linear combination. Thus, given a cycle in an undirected graph, *every* orientation which constitutes a simple cycle has a $\{-1, 0, +1\}$ linear combination, as well.

For the non-implications "G-TUM $\not\Longrightarrow$ S-TUM" and "C-TUM $\not\Longrightarrow$ S-TUM" look at Example 5.22 with a graph and a cycle basis which is C-TUM and G-TUM, but not S-TUM.

Example 5.22.
The considered graph in this example is a directed version of the sunflower graph SF(3), see Figure 5.3.

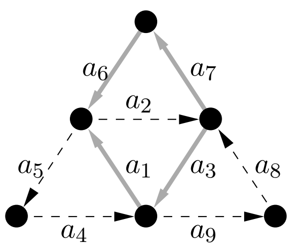


Figure 5.3: The sunflower graph SF(3) with an orientation and the circuits C_5 (grey arcs) and C_6 (dashed arcs).

The cycle basis B contains the four triangles

$$C_1 = (1, 1, 1, 0, 0, 0, 0, 0, 0)^\mathsf{T}, \qquad C_2 = (1, 0, 0, 1, 1, 0, 0, 0, 0)^\mathsf{T},$$
$$C_3 = (0, 1, 0, 0, 0, 1, 1, 0, 0)^\mathsf{T}, \text{ and } \quad C_4 = (0, 0, 1, 0, 0, 0, 0, 1, 1)^\mathsf{T}.$$

Now, we show that this basis is C-TUM and G-TUM, but not S-TUM.

C-TUM. Due to symmetry, it is sufficient to examine only the circuits

$$C_5 = (1,0,1,0,0,-1,-1,0,0)^\mathsf{T}, \quad C_6 = (0,-1,0,1,1,0,0,1,1)^\mathsf{T}, \text{ and}$$
$$C_7 = (0,0,0,1,1,1,1,1,1)^\mathsf{T}.$$

The circuit C_5 is depicted by grey arcs in Figure 5.3 while C_6 is drawn with dashed arcs. C_7 is the circuit which is constituted by the boundary of the unbounded face. These circuits have the representations $C_5 = C_1 - C_3$, $C_6 = -C_1 + C_2 + C_4$, and $C_7 = -C_1 + C_2 + C_3 + C_4$. This shows that the basis B is C-TUM.

G-TUM. As shown in the paragraph above, all circuits have a $\{-1, 0, +1\}$ representation. The only non-circuit in the underlying graph $G(\mathrm{SF}(3))$ is the one which contains all edges. For this cycle, the orientation $C_8 = (1,1,1,1,1,1,1,1,1)^\mathsf{T}$ admits the linear combination $C_8 = C_2 + C_3 + C_4$.

Not S-TUM. The cycle $C_9 = (-1,-1,-1,1,1,1,1,1,1)^\mathsf{T} = -2C_1 + C_2 + C_3 + C_4$ does not come up with a representation as demanded for an S-TUM cycle basis. ♢

With this example, we have shown that S-TUM cycle bases are a proper subclass of G-TUM respectively of Γ-TUM cycle bases. This, in turn, solves one part of Open Problem 3 in [64]. On the other hand, S-TUM cycle bases are also a proper subclass of C-TUM cycle bases. This reinforces the assumption that the notions of C-TUM cycle bases and Γ-TUM cycle bases are equivalent—the other part of Open Problem 3.

By taking a look back at the diagram, it can be observed that also strictly fundamental cycle bases are a proper subclass of C-TUM cycle bases. Otherwise, C-TUM and S-TUM cycle bases would be the same. However, since the cycle basis in Example 5.22 is not strictly fundamental, it also serves as a separating example for C-TUM and S-TUM bases.

It is still open whether Γ-TUM cycle bases are a proper subclass of C-TUM cycle bases on one hand, and whether strictly fundamental cycle bases are a proper subclass of S-TUM cycle bases.

As of now, Γ-TUM respectively G-TUM cycle bases are denoted by *totally unimodular cycle bases* or *TUM bases*.

5.2.3 TUM Bases vs. Weakly Fundamental Cycle Bases

This subsection is dedicated to compare totally unimodular with weakly fundamental cycle bases. This had already been done in [78], where the author presented a minimum cycle basis and claimed it to be totally unimodular but not weakly fundamental (Example 11.6). We will show that the given cycle basis is *not* totally unimodular. This is shown in the first part of this subsection. In the second part, we present a new example of a cycle basis which is indeed not weakly fundamental but totally unimodular. This basis is not the minimum one of its graph. The reason is that it contains non-circuits. Due to the Exchange Theorem 3.49, such a basis cannot be minimum. On the other hand, Such a non-circuit cannot simply be replaced by a part of it while preserving totally unimodularity. We call this property the *Exchange Property* and formulate it as a stronger form of the Exchange Theorem. Actually, this holds for weakly fundamental and for undirected cycle bases. After an example which shows that the Exchange Property does not hold for TUM bases, we proof that under additional constraints each totally unimodular cycle basis which consists only of circuits is also weakly fundamental.

Example 11.7 in [78]. We show that the cycle basis in this example is not totally unimodular. The graph considered there consists of six copies of the graph in Figure 5.4 and of the indicated cycle basis.

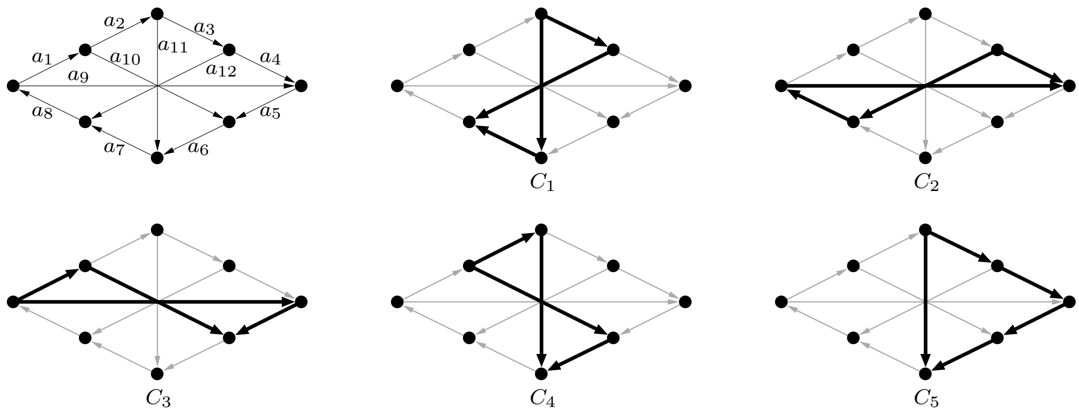


Figure 5.4: Wagner's graph and a cycle basis.

The according cycle matrix has the form

$$\Gamma = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & \mathit{0} & \mathit{0} & \mathit{-1} & \mathit{1} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & \mathit{-1} & \mathit{0} & \mathit{0} & \mathit{-1} \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \mathit{1} & \mathit{-1} & \mathit{0} & \mathit{0} \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & \mathit{0} & \mathit{1} & \mathit{-1} & \mathit{0} \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}^{\mathsf{T}},$$

which is not totally unimodular since the emphasized submatrix has the determinant 2. Due to Proposition 5.15, the cycle basis of the graph which is composed by the six copies is not totally unimodular, as well.

A TUM Basis Which is not Weakly Fundamental. Example 5.23 presents a graph D with a cycle basis B which is totally unimodular but neither weakly fundamental nor minimum.

Example 5.23.
Consider the graph in Figure 5.5 and the indicated cycle basis.

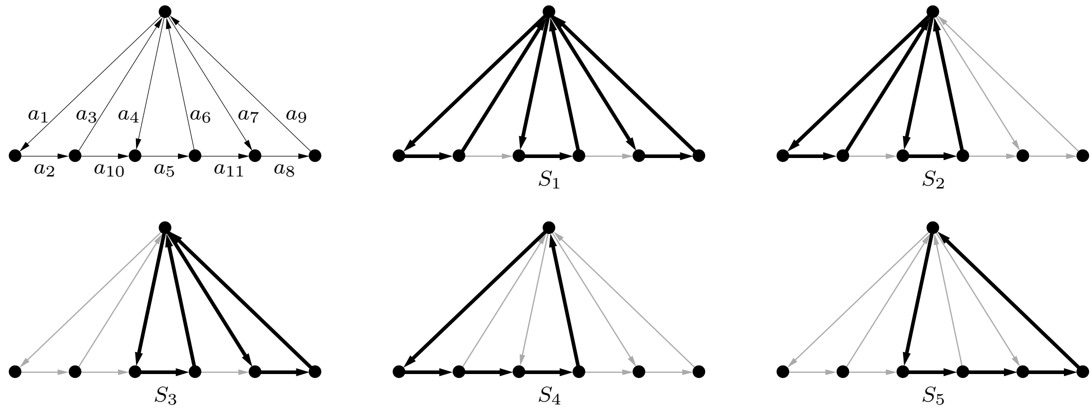


Figure 5.5: A graph with a totally unimodular cycle basis which is not weakly fundamental.

It can easily be checked that the cycle basis is not weakly fundamental. After removing the circuits S_4 and S_5 from B and the arcs a_{10} and a_{11} from D, each remaining arc is contained in at least two of the cycles left over. The cycle matrix is

$$\Gamma = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}^{\mathsf{T}}.$$

After removing double columns and columns with only one non-zero element according to Lemma 5.17, this matrix reduces to

$$\Gamma' = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}}$$

To see that this matrix is totally unimodular, we could investigate all $2^5 = 32$ subsets of the rows if there is a split according to Lemma 5.16. Clearly, this can be done more economically since the empty set as well as the one element subsets are not of interest.

Moreover, there is a split for each two element subsets because the matrix does not contain a -1 entry. Each entry of the first row is one, thus, subtracting two rows from it leads to a vector as demanded. So we do not have to consider all three element subsets containing the first row. For the other three element subsets observe that the 1 entries in the forth row are a subset of the entries in the second row. The same does hold for the fifth and the third row. Because each remaining three element subset contains either the second and the fourth, or the third and the fifth row, we do not have to test these combinations, either.

For the 5 four element subsets and for the entire set we provide the splits below. We denote by Γ'_i the ith row of Γ'.

$$\begin{aligned}
\Gamma'_1 - \Gamma'_2 - \Gamma'_3 + \Gamma'_4 &= (1, 0, -1, 0, 0, 0, 0), \\
\Gamma'_1 - \Gamma'_2 - \Gamma'_3 + \Gamma'_5 &= (0, 0, 0, 0, -1, 0, 1), \\
\Gamma'_1 - \Gamma'_2 + \Gamma'_4 - \Gamma'_5 &= (1, 0, -1, 0, 1, 1, 0), \\
\Gamma'_1 - \Gamma'_3 - \Gamma'_4 + \Gamma'_5 &= (0, 1, 1, 0, -1, 0, 1), \\
\Gamma'_2 + \Gamma'_3 - \Gamma'_4 - \Gamma'_5 &= (0, 1, 1, 0, 1, 1, 0), \quad \text{and} \\
\Gamma'_1 - \Gamma'_2 - \Gamma'_3 + \Gamma'_4 + \Gamma'_5 &= (1, 0, 0, 1, 0, 0, 1).
\end{aligned}$$

This shows that Γ' as well as Γ itself is totally unimodular. $\Diamond$

A TUM Basis Without the Exchange Property. Clearly, this basis is not minimum as required for Example 11.6 in [78]. Given that a cycle basis B is weakly fundamental and that B contains a non-circuit, a straightforward way to construct out of B a weakly fundamental cycle basis with a smaller weight is stated in the next lemma.

Lemma 5.24 (Exchange Property, cf. Lemma 3.12 in [64]). *Let B be a weakly fundamental cycle basis of a directed graph D, $S \in B$ a simple cycle that is not a circuit and $S = S_1 + S_2$ with $\operatorname{supp}(S_i) \subset \operatorname{supp}(S)$ for $i = 1, 2$. Then at least one of $B \setminus \{S\} \cup \{S_1\}$ or $B \setminus \{S\} \cup \{S_2\}$ is a weakly fundamental cycle basis of D.* $\square$

The Exchange Property should not be confused with the Exchange Theorem 3.49. In the Exchange Theorem, the condition $\operatorname{supp}(S_i) \subset \operatorname{supp}(S)$ for $i = 1, 2$ is dropped. The only subclass of directed cycle bases for which the Exchange Theorem is valid are undirected cycle bases.

Lemma 5.24 implies that each minimum weakly fundamental cycle basis contains only circuits. Obviously, this is also true for strictly fundamental cycle bases. Additionally, it does also hold for undirected cycle bases, see Theorem 3.14 in [64]. The question arises, whether this is also true for integral or totally unimodular cycle bases. This is one part of Open Problem 4 in the Survey. Another part of this problem is the question whether

Lemma 5.24 does hold for totally unimodular cycle bases. In Example 5.25, we present a cycle basis which shows that Lemma 5.24 is not true for totally unimodular cycle bases.

Example 5.25.
The totally unimodular cycle basis in this example contains a non-circuit S_1 which can be decomposed into the circuits S_4 and $C = (0,0,0,1,1,1,1,0,0,0)^\mathsf{T}$, see Figure 5.6. The cycle matrix loses its totally unimodularity if S_1 is replaced by C. On the other hand, it is not a basis anymore after the replacement of S_1 with S_4.

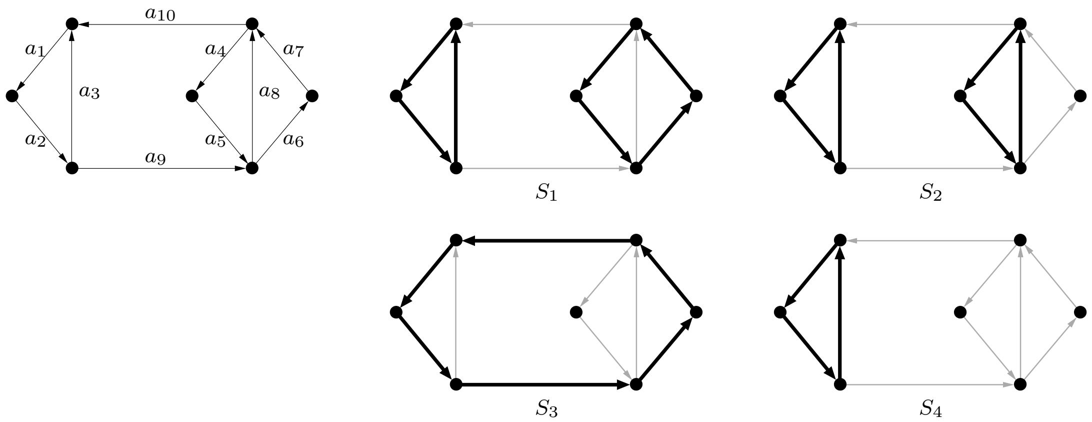


Figure 5.6: Example of a totally unimodular cycle basis for which Lemma 5.24 does not hold.

The cycle matrix of B is

$$\Gamma = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T}.$$

With a similar argumentation as in Example 5.23, it is sufficient to find splits for the last three basic cycles and for the set of all basic cycles as well. These are given by $\Gamma_2+\Gamma_3-\Gamma_4 = (1,1,0,1,1,1,1,0,1,1)^\mathsf{T}$ and by $\Gamma_1-\Gamma_2-\Gamma_3+\Gamma_4 = (0,0,1,0,0,0,0,0,-1,-1)^\mathsf{T}$. It is easy to see that $B \setminus \{S_1\} + \{S_4\}$ is not a cycle basis. The cycle matrix for $B \setminus \{S_1\} + \{C\}$ is

$$\Gamma' = \begin{pmatrix} \mathit{0} & 0 & 0 & \mathit{1} & 1 & \mathit{1} & 1 & 0 & 0 & 0 \\ \mathit{1} & 1 & 1 & \mathit{1} & 1 & \mathit{0} & 0 & 0 & 0 & 0 \\ \mathit{1} & 1 & 0 & \mathit{0} & 0 & \mathit{1} & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T},$$

where the emphasized submatrix has determinant -2. $\Diamond$

Restriction to Bases Consisting of Circuits. In this paragraph, we make further restrictions on the cycles in the cycle basis and on the graph itself. We show that with respect to these restrictions, each totally unimodular cycle basis is also weakly fundamental. More precisely, we consider only biconnected graphs. For a cycle basis B, we require the following:

1. B consists only of circuits,
2. each arc is contained in at least two basic circuits, and
3. no arc is contained in more than three basic circuits.

Item 1 is necessary due to Example 5.23. Item 2 can be assumed without loss of generality. If there were arcs which are contained in only one basic circuit, then our argumentation below will also work without these arcs and the according basic circuits. For our proof, we also need Item 3. However, we conjecture that Item 3 is not necessary for the statement. We proceed with two technical lemmas which involve the restrictions above.

Lemma 5.26. *If a cycle basis B of a directed or undirected graph D is not weakly fundamental, each arc of D is contained in at least two circuits of B, and B does not contain any non-circuit, then for each circuit C in B, there are two circuits $C_1 \neq C_2$ in B with $C_1 \neq C \neq C_2$ and $C_1 \cap C \neq \emptyset \neq C_2 \cap C$.*

Proof. Assume that there were a circuit $C \in B$ for which Lemma 5.26 does not hold. Since each arc is contained in two basic circuits, each arc of C must be in another element $C_1 \in B$. However, if C_1 were the only such circuit in B and $C_1 \neq C$, then C_1 cannot be a circuit. Thus, there must be a second circuit C_2 as stated in the lemma. ☐

Definition 5.27 (circuit chain, chain link, (un-)twisted closed circuit chain). *Let B be a cycle basis of a digraph. A* circuit chain *is a sequence $S(1, k) = (C_1, \ldots, C_k)$ of pairwise distinct basic circuits of B with $C_{i-1} \cap C_i \neq \emptyset$ for $i \in 2, \ldots, k$. An element in $C_{i-1} \cap C_i$ is called* chain link. *A circuit chain is* closed *if $C_1 = C_k$. If the circuits of a circuit chain can be orientated such that in their sum, all chain links cancel out each other, then the circuit chain is called* untwisted. *Otherwise, it is a* twisted circuit chain*; it is twisted like a Möbius stripe. Cf. Figure 5.7 for a twisted and an untwisted closed circuit chain.*

Lemma 5.28. *Let D be a biconnected digraph and B a cycle basis of D which contains only circuits and each arc of D is contained in at least two basic circuits. Then for each two basic circuits $C_1, C_2 \in B$ there is a circuit chain $S(1, 2) = (C_1, \ldots, C_2)$.*

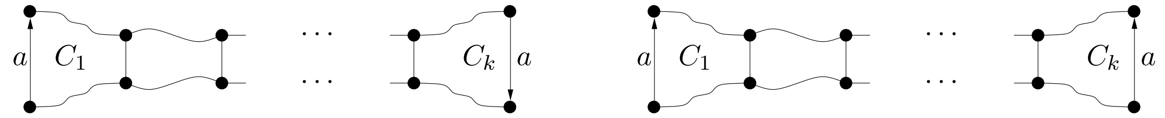


Figure 5.7: A twisted (a) and an untwisted (b) circuit chain.

Proof. Assume that there were two circuits C_1 and C_2 for which such a circuit chain does not exist. Then there exist two disjoint sets $B_1, B_2 \subset B$ with $C_1 \in B_1$ and $C_2 \in B_2$ so that for each two circuits $C_i', C_i'' \in B_i$, there is a circuit chain $(C_i', \ldots, C_i'')$ for $i \in \{1,2\}$. These sets induce two subgraphs D_1 and D_2 of D. Since D is biconnected, it follows from Menger's Theorem (Theorem 5.7) that there is a circuit C with two vertices $v_1 \in C \cap V(D_1)$ and $v_2 \in C \cap V(D_2)$. This circuit C intersects in a path with D_1 and in a path with D_2, respectively. Clearly, these paths cannot be represented only with basic circuits in one of the sets B_1 and B_2. Because B is a cycle basis of D, the only possibility of representing C is that there is a circuit $C_3 \in B$ with $C_3 \cap V(D_1) \neq \emptyset \neq \emptyset C_3 \cap V(D_2)$. This means that there are circuit chains $S(1,3) = (C_1, \ldots, C_3)$ and $S(2,3) = (C_2, \ldots, C_3)$. But then also the chain $S(1,2) = (C_1, \ldots, C_2)$ exists. $\square$

In what follows, we have been inspired by the proof of the second claim in Example 11.7 in [78]. More precisely, we conjecture that one can always find a circuit chain such that the characterization of totally unimodular cycle bases given in Lemma 5.16 is not fulfilled if the cycle basis is not weakly fundamental. Additionally, we will profit from the fact that for a non-fundamental cycle basis, there is an arc that is contained in three basic circuits, cf. Lemma 3.2.

Theorem 5.29. *Let D be a biconnected digraph and B an undirected cycle basis which contains only circuits and each arc of D is contained in at most three basic circuits. If B is totally unimodular then it is also weakly fundamental.*

Proof. Let $a = uv$ be an arc which is contained in three basic circuits C_1, C_2 and C_3 from the non-fundamental undirected cycle basis B. We may assume the existence of such an arc, since otherwise, according to Lemma 3.2, the cycle bases is immediately weakly fundamental. For the sake of clarity, we may embed the arc a twice, similarly to the embedding of Champetier's graph in Example 11.7 in [78] or in Example 4.17 in this thesis. Moreover, we also may choose the embedding of the circuits C_1, C_2 and C_3 arbitrarily. Our preferred embeddings are illustrated in the next figures.

Without loss of generality assume that $C_3 \notin S(1,2)$. Otherwise, the roles of C_2 and C_3 can be interchanged. Analogously, we may assume $C_1 \notin S(2,3)$. We distinguish whether there are twisted circuit chains between any two of the three circuits C_1, C_2 and C_3 or not.

Case 1 (Twisted circuit chain between C_1 and C_2).
Without loss of generality, we may assume that the circuit chain $S(1,2)$ is twisted, i.e. the

situation in Figure 5.8. At first, suppose that each chain link is contained in exactly two circuits of $S(1,2)$. Choose an arbitrary orientation of C_1, say, counterclockwise such that a is oriented in the direction of the arrow. Then, each subsequent circuit in the chain also has to be oriented counterclockwise. Otherwise, the entry according to the chain link at the changeover from counterclockwise to clockwise orientation is -2 or $+2$. In particular, also C_2 has to be oriented counterclockwise. But then, the value associated to a is 2.

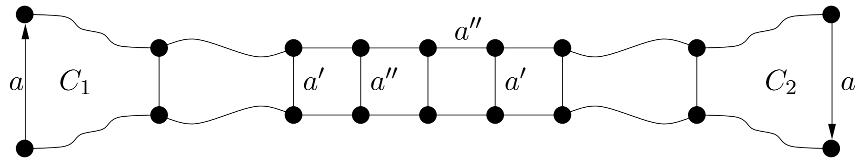


Figure 5.8: The arc a, the circuits C_1 and C_2, an indicated circuit chain $S(1,2)$ from C_1 to C_2. In addition, some chain links a' and a'' with the properties described in the text are marked.

Suppose now that there is a chain link that occurs in a third circuit of the circuit chain. There are essentially two types of such chain links. A chain link can serve as a chain link again, for example a' in Figure 5.8, or not, as the chain link a'' in this figure. In both cases, the circuit chain can be abbreviated by cutting out the part between the first and the last appearance of such a circuit link. Dependent on the orientation of the last copy of the circuit link, we have to consider this case or the next one.

Case 2 (No twisted circuit chain between C_1 and C_2).
Also in this case we can make the same assumptions as above, i.e. $C_3 \notin S(1,2)$, $C_1 \notin S(2,3)$, and for both circuit chains, each chain link is contained in exactly two circuits of them. Similarly to Figure 5.8, we embed two copies of the arc a. This time, both circuit chains, $S(1,2)$ and $S(2,3)$ are depicted, where there are also two copies of the circuit C_2. See Figure 5.9 for our embedding.

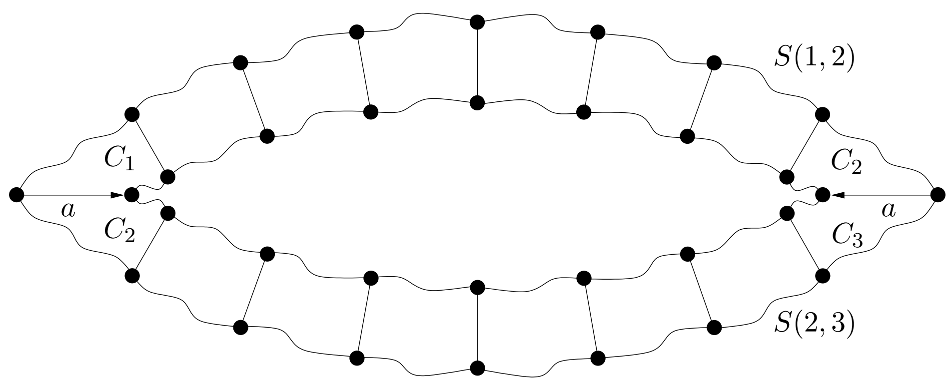


Figure 5.9: The other relevant possibility of embedding the circuit chains $S(1,2)$ and $S(2,3)$.

Arcs which are no chain links induce paths, where several of these paths also can be empty. But note that not all of these paths can be empty, otherwise, the cyclomatic number of the embedded graph would be smaller than the number of the circuits in the union of the circuit chains. Thus, we can choose an arc a_1 in a path of a basic circuit from $S(1,2)$ which is not in another circuit from both circuit chains, but in a further circuit C_1'. Another arc a_2 with this property can be chosen from $S(2,3)$, this arc is in a circuit C_2'. Between C_1' and C_2' there is a circuit chain, see Figure 5.10.

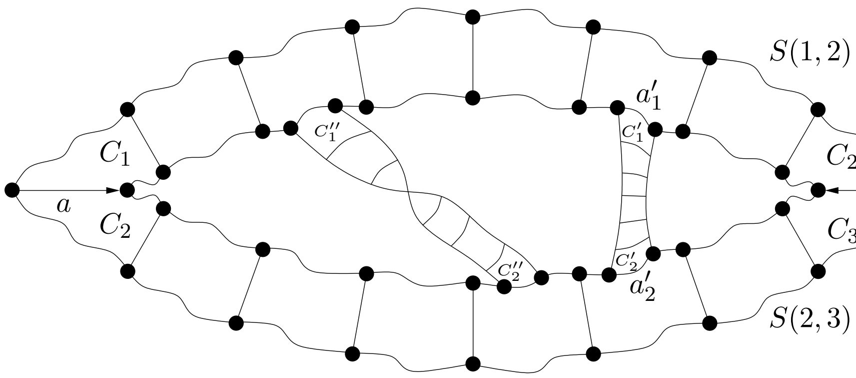


Figure 5.10: Twisted and untwisted circuit chain in the embedding according to the setting of the proof.

This circuit chain must not be twisted like the circuit chain between C_1'' and C_2'', because this circuit chain and the chains from C_1 to C_1'' and from C_2'' to C_2 would induce a twisted circuit chain from C_1 to C_2, a contradiction to our assumption for this case. Hence, it must have the form as from C_1' to C_2'. This chain together with the chains from C_1 to C_1' and from C_2' to C_3 induces a twisted circuit chain from C_1 to C_3. The circuits in this chain do not provide a split as demanded in Lemma 5.16, the reason is the same as already described at the beginning of Case 1. ☐

We conjecture that Theorem 5.29 also holds if there are arcs which are contained in more than three basic circuits.

Conjecture 5.30. *Let D be a biconnected digraph and B an undirected cycle basis which contains only circuits. If B is totally unimodular then it is also weakly fundamental.*

5.2.4 Conclusions

In this section, we dealt with totally unimodular cycle bases. We shed a little bit light on the issue of the different notions of TUM bases. It remains open how Γ-TUM cycle bases and C-TUM cycle bases on one hand, and S-TUM cycle bases and strictly fundamental cycle bases on the other hand relate to each other.

Another topic was the relationship between totally unimodular cycle bases and weakly fundamental cycle bases. Broadened to cycle bases that contain non-circuits and without the requirement of minimality, we were able to repair the map on the hierarchy of classes of cycle bases given in [78]. In contrast, if only circuits are allowed, then this map has another appearance when we make further restrictions on the cycle basis. The question at this place is whether this result is also valid without these restrictions.

5.3 Integral Cycle Bases

The matter of this section is integral cycle bases. In contrast to the other two classes presented in this chapter, integral cycle bases are of considerable practical interest. This is indicated in Subsection 5.3.1, which is based on the introduction of [108]. Subsection 5.3.2 describes the PERIODIC EVENT SCHEDULING Problem and its relationship to integral cycle bases in detail. In the following section, we define integral cycle bases and give the proof of its equivalence to two further imaginable versions in Subsection 5.3.3. The last two subsection contain results on integral cycle bases without simple cycles and on the coefficients of the linear combinations of circuits.

5.3.1 Introduction

The concept of integral cycle bases has been introduced by Liebchen and Peeters in [79] (see also [77] and [80]) as a generalization of strictly fundamental cycle bases. In their paper, the authors used integral cycle bases to characterize periodic tensions which appear in a model for cyclic timetables.

Cyclic timetables are constructed to describe periodically recurring activities or events. There are numerous advantages of periodicity compared to aperiodic processes. Some of them are fairly banal, like convenience for the users. An important technical advantage is the efficiency, because it is sufficient to allegorize only one period. Hence, periodically recurring events can be presented in a much more compact way. This can result in smaller complexity and a better clarity over the supply, what can lead to a considerable saving of time.

There are different ways of constructing a cyclic timetable. One option is to use existing techniques for computing non-periodic timetables. By adding some extra constraints, these techniques can constitute a periodic timetable. This method is known as *synchronizing individually scheduled trips*. Another approach is to use the QUADRATIC SEMI-ASSIGNMENT Problem. For a more detailed description of these two methods and related literature we refer to [78].

It emerged that the PERIODIC EVENT SCHEDULING Problem (PESP for short) is a much more promising approach for modeling periodic timetables. This problem had been introduced in 1989 by Serafini and Ukovich in [114]. They also could prove the $\mathcal{NP}$-completeness of the PESP.

In the next subsection, we explain why integral cycle bases are useful for modeling periodic timetables.

5.3.2 The PESP for Modeling Periodic Timetables

This subsection is dedicated to describe and to formulate the PESP and its relationship to integral cycle bases. For further applications of PESP and its "modeling power" we refer to Chapter 7 of [78]. These references also emphasizes the importance of integral cycle bases. The subsection is based on the descriptions of PESP in the PhD theses of Peeters [100] and Liebchen [78] as well as on their technical report [79]. A similar description can also be found in [108].

We start to describe the model of PESP by using notations from the railway. An *event* in this model is illustrated by a triplet (train, station, departure) or (train, station, arrival). Here, departure and arrival are interpreted as integer points in time within a *period length* T. The set of these events is denoted by V, which is later interpreted as the vertex set of a directed graph. The function $\pi : V \to \{0, 1, \ldots, T-1\}$ assigns each event to a moment within the period.

For several pairs of events, the time difference between them is constrained. For instance, this is done to avoid long waiting times (upper bounds) or to enable acceptable interchange facilities (lower bounds). Formally, the difference between both events in such a pair $a = (i, j)$ should lie in a time interval $[\ell_a, u_a]$, that is:

$$\pi_j - \pi_i \in [\ell_a, u_a]. \tag{5.3}$$

Since each event recurs periodically, an integer multiple of T is added to the difference:

$$\pi_j - \pi_i + Tp_a \in [\ell_a, u_a], \quad p_a \in \mathbb{Z}. \tag{5.4}$$

Note that p_a can also be negative. Furthermore, we make the following restrictions. Both, the lower and the upper bound of the time interval should be non-negative integers. Moreover, it is sufficient that the lower bound is chosen to be at most $T-1$. The upper bound has to be larger than the lower, and it must not be at a later point in time than the according lower bound of the next period. Hence, we get the two inequalities $0 \le \ell_a \le T-1$ and $\ell_a \le u_a \le T + \ell_a$. Putting both inequalities together, we obtain

$$0 \le u_a - \ell_a < T. \tag{5.5}$$

With A being the set of event pairs, the digraph $D = (V, A)$, the period length T, and the vectors ℓ and u describe an instance of the PESP. To solve this problem, one has to find the function π and a vector p with the entries p_a from (5.4). Formally, the PESP is the following problem:

PERIODIC EVENT SCHEDULING PROBLEM (PESP)			
Instance:	Digraph	$D = (V, A)$,	
	period length	T,	
	vectors	ℓ and u.	
Question:	Is there a solution	(π, p)	
	subject to	$\pi_j - \pi_i + Tp_a \in [\ell_a, u_a]$	$\forall\, a = (i,j) \in A$
		$0 \le \pi_i < T$	$\forall\, i \in V$
		$\pi_i \in \mathbb{Z}$	$\forall\, i \in V$
		$p_a \in \mathbb{Z}$	$\forall\, a = (i,j) \in A$?

Considering π as a *potential*, a function $x : A \to \mathbb{Z}$ is said to be a *periodic tension with period T* if it has the form $x_a = \pi_j - \pi_i + Tp_a$ for all $a = (i,j) \in A$, a potential π, and an integer vector p.

A simple cycle S of D has the *cycle periodicity property* (CPP) if for some $q_S \in \mathbb{Z}$ the following holds

$$\sum_{a \in S^+} x_a - \sum_{a \in S^-} x_a = Tq_S. \tag{5.6}$$

The CPP can easily be extended to integral cycles. Clearly, one has to take into account how often an arc occurs in the integral cycle. Therefore, we suggest

$$\sum_{a \in I} I(a)x_a = Tq_I \tag{5.7}$$

as a generalization of the CPP that can also be applied to integral cycles. As $S(a)$ is in $\{-1, 0, +1\}$ for each arc in a simple cycle S, Equality (5.7) is indeed a generalization of Equality (5.6). A further generalization to circulations would not make sense, since T and q_S have to be integer in this model.

In this context, Nachtigall observed in [92] the following relationship between (5.6) and periodic tensions.

Theorem 5.31. *For a connected digraph $D = (V, A)$ and a period T, a function $x : A \to \mathbb{R}$ is a periodic tension with period T if and only if each circuit has the cycle periodicity property (5.6).* $\square$

The only thing that is still left to associate a periodic tension with a solution of the PESP is the fulfilment of the time slots for each value x_a. Requiring each q_C in (5.6) to be integer, we obtain the CYCLE PERIODICITY FORMULATION (CPF):

CYCLE PERIODICITY FORMULATION (CPF)			
Instance:	Digraph	$D = (V, A)$,	
	period length	T,	
	vectors	ℓ and u.	
Question:	Is there a solution	(x, q)	
	subject to	$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = Tq_C$	for each circuit $C \in D$
		$\ell_a \le x_a \le u_a$	for all $a \in A$
		$x_a \in \mathbb{R}$	for all $a \in A$
		$q \in \mathbb{Z}^C$?	

The authors of [79] did not explicitly distinguish between the terms *circuit* and *simple cycle*. In our version of the CPF, we decided to use circuits. However, since a simple cycle is the direct sum of circuits both formulations are equivalent.

The proof of Theorem 5.31 in [79] yields an instruction for constructing a solution for the PESP from a solution of the CPF. But to find a solution of the CPF, one possibly has to consider exponentially many circuits. Inspired by the fact that for an a-periodic tension it suffices to require $\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = 0$ for the circuits of a cycle basis, the authors made a similar observation.

Theorem 5.32 (Theorem 3.2 in [79]). *If the cycle periodicity property (5.6) holds for every circuit in an integral cycle basis of a graph D, then it holds for every circuit in D.* □

5.3.3 Definition

This subsection contains the definition of integral cycle bases. Two further versions for a definition are also possible. Anyway, we show that all three versions are equivalent.

Definition 5.33 (integral cycle basis). *A directed cycle basis $B = \{C_1, \ldots, C_\nu\}$ of a digraph D is an* integral cycle basis *if each circuit C of D has a representation $C = \sum_{i=1}^{\nu} \lambda_i C_i$, where all λ_i are integers.*

We also could have required that each simple cycle or each integral cycle has an integer representation of basic circuits. Actually, the three possible definitions are equivalent. Since each circuit is a simple cycle and each simple cycle is an integral cycle, we have only to show the following implication.

Lemma 5.34. *Let B be an integral cycle basis of a graph $D = (V, A)$ as defined in Definition 5.33. Then every integral cycle in D has an integer representation of basic circuits.*

Proof. Let I be an arbitrary integral cycle in D. We show that the representation of I as the sum of basic circuits has integral coefficients. Therefore, replace D with the auxiliary graph $D' = (V, A')$, where A contains $|I(a)|$ copies of the arc a. It follows that $\deg_I^+(v) = \deg_I^-(v)$ holds for each $v \in V$ and that I has a decomposition into circuits of D', i.e. $I = \sum_{j=1}^{k} C'_j$ for a certain k, see, for example [18]. Regarding these circuits as circuits in D, this equality holds also in D. Since each C'_j is a circuit, it has an integer representation, namely $C'_j = \sum_{i=1}^{\nu} \lambda_i^j C_i$ with $C_i \in B$ and $\lambda_i^j \in \mathbb{Z}$ for all $i \in \{1, \ldots, \nu\}$ and $j \in \{1, \ldots, k\}$. Then

$$I = \sum_{j=1}^{k} C'_j = \sum_{j=1}^{k} \sum_{i=1}^{\nu} \lambda_i^j C_i = \sum_{i=1}^{\nu} \left[\left(\sum_{j=1}^{k} \lambda_i^j \right) C_i \right]$$

is the integer linear combination of basic circuits for I. □

5.3.4 An Integral Cycle Basis Without Simple Cycles

Since each circuit must have an integer representation, one may think that every element in an integral cycle basis has to be a circuit or, at least, a simple cycle, i.e. for each $I \in B$ we have $|I(a)| = 1$ for all $a \in I$. But, surprisingly, this is not necessary. We present a graph D and an infinite family $B^\ell = \{I_1^\ell, \ldots, I_\nu^\ell\}$ of integral cycle bases where for each $n_0 \in \mathbb{N}$ there is an ℓ such that for all $i \in \{1, \ldots, \nu\}$ we have $|I_i^\ell(a)| > n_0$ for all $a \in I_i$.

Example 5.35.
Consider the small graph depicted in Figure 5.11 and the indicated cycle basis, which does not contain a simple cycle.

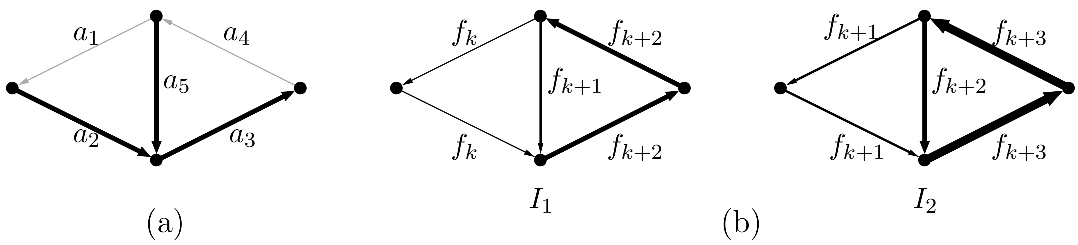


Figure 5.11: A graph with a spanning tree (a) and an integral cycle basis that does not contain a simple cycle (b). The widths of the arcs represent the value of the entry in the vector which corresponds to the cycle.

With f_k being the kth Fibonacci number, the cycle matrix is

$$\Gamma = \begin{pmatrix} f_k & f_k & f_{k+2} & f_{k+2} & f_{k+1} \\ f_{k+1} & f_{k+1} & f_{k+3} & f_{k+3} & f_{k+2} \end{pmatrix}^{\mathsf{T}}.$$

Delete the second, the third, and the fifth column, i.e. the arcs corresponding to the spanning tree in the left graph in Figure 5.11. The obtained submatrix Γ' has determinant

$\det(\Gamma') = |f_k f_{k+3} - f_{k+1} f_{k+2}| = 1$, what shows that the pair of the indicated integral cycles actually forms an integral cycle basis.

The value of the determinant in the context of Fibonacci numbers is a direct consequence of d'Ocagne's indentity for Fibonacci numbers, namely $f_{k_1} f_{k_2+1} - f_{k_2} f_{k_1+1} = (-1)^{k_2} f_{k_1-k_2}$. $\diamondsuit$

5.3.5 TUM Bases and the Exchange Property

As already mentioned in Section 5.2, it is unknown whether Lemma 5.24, the Exchange Property, is valid for integral cycle bases. In this subsection, we prove a weaker statement. Moreover, we present an example which shows that the proof of this weaker statement cannot be carried over to a proof of the Exchange Property for integral cycle bases in a straightforward way.

Lemma 5.36. *Let B be a totally unimodular cycle basis containing a non-circuit. Then there is an integral cycle basis B' with $w(B') < w(B)$.*

Proof. This proof uses the same decomposition of an integral cycle that we have already seen in the proof of Lemma 5.34. Without loss of generality let I_1 be a non-circuit of the totally unimodular cycle basis $B = \{I_1, \ldots, I_\nu\}$. Then it has a decomposition $I_1 = \sum_{j=1}^{k} C_j$ into circuits. Each of these circuits has a representation $C_j = \sum_{i=1}^{\nu} \lambda_i^j I_i$ with $\lambda_i^j \in \{-1, 0, +1\}$, and therefore, we have $I_1 = \sum_{j=1}^{k} \sum_{i=1}^{\nu} \lambda_i^j I_i$. There must be at least one index j for which

$$|\lambda_1^j| = 1, \tag{5.8}$$

otherwise, I_1 would be linearly dependent from $\{I_2, \ldots, I_\nu\}$. Without loss of generality let $|\lambda_1^1| = 1$, then we can replace I_1 by C_1 and since $C_j \subset I_1$ for all j, we have $w(C_1) < w(I_1)$. It remains to show that $\{C_1, I_2, \ldots, I_\nu\}$ is an integral basis. Therefore, consider an arbitrary integral cycle $I = \sum_{i=1}^{\nu} \lambda_i I_i$. We have

$$I_1 = -\frac{1}{\lambda_1^1} C_1 + \sum_{i=2}^{\nu} \frac{\lambda_i^1}{\lambda_1^1} I_i \tag{5.9}$$

and therefore after the replacement

$$I = -\frac{\lambda_1}{\lambda_1^1} C_1 + \sum_{i=2}^{\nu} \left(\frac{\lambda_1 \lambda_i^1}{\lambda_1^1} + \lambda_i \right) I_i \ , \tag{5.10}$$

which is an integer representation of I, since $|\lambda_1^1| = 1$. If this procedure is repeated for each non-circuit in the totally unimodular cycle basis, we obtain an integral cycle basis which entirely consists of circuits and which has a smaller weight. $\square$

This proof does not work if B is an integral cycle basis. More precisely, it cannot be presumed that one can always find an index for which Equation (5.8) actually holds. Example 5.37 shows a digraph D, an integral cycle basis $B = \{C_1, \ldots, C_\nu\}$ of D, and a circuit $C = \sum_{i=1}^{\nu} \lambda_i C_i$ in D such that $|\lambda_i| \neq 1 \; \forall i = 1, \ldots, \nu$.

Example 5.37.
Consider the graph D depicted in Figure 5.12. It is formed by the two generalized Petersen graphs $P_{7,3}$ and $P_{7,2}$ after identifying the copies of i_j for $j = 0, \ldots, 6$.

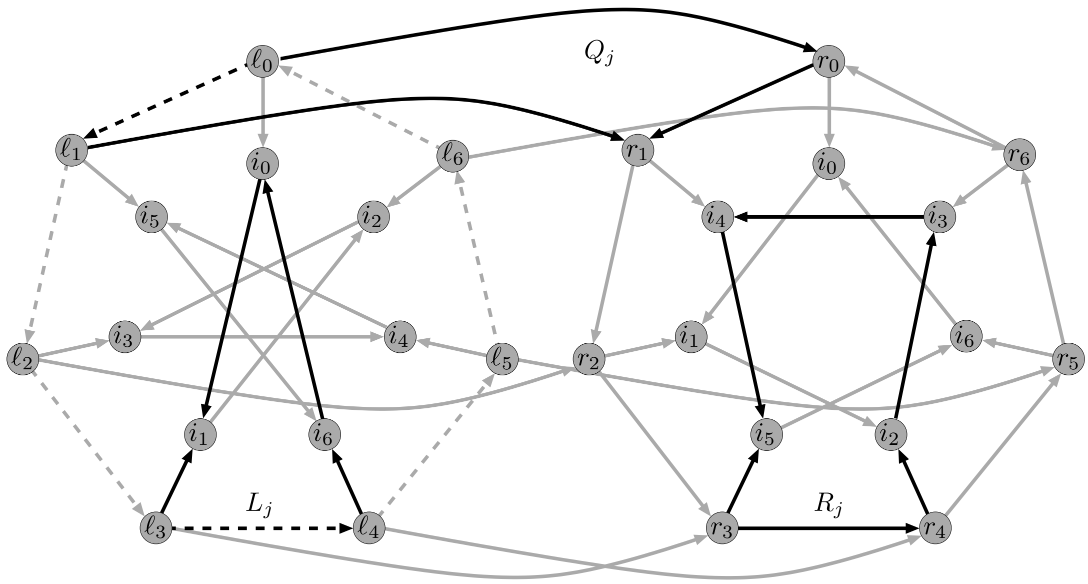


Figure 5.12: A graph D with an integral cycle basis (circuits L_j, R_j and Q_j, fat arcs) and a circuit that contains each basic circuit at least twice in its representation (dashed arcs).

D consists of 21 nodes and 42 arcs, where the arcs are partitioned into the following six groups. In what follows, all indices are modulo 7.

Group	Notation
left outer arc	$\ell_j \ell_{j+1}$
right outer arc	$r_j r_{j+1}$
inner arc	$i_j i_{j+1}$
left spoke	$\ell_j i_{j'}$
right spoke	$r_j i_{j'}$
cross spoke	$\ell_j r_j$

The cycle basis B contains *left circuits* $L_j = \{\ell_j\ell_{j+1}, \ell_{j+1}i_{5-2j}, i_{5-2j}i_{6-2j}, i_{6-2j}i_{-2j}, \ell_j i_{-2j}\}$, *right circuits* $R_j = \{r_j r_{j+1}, r_{j+1}i_{-3j}, i_{-3j}i_{1-3j}, i_{1-3j}i_{2-3j}, i_{2-3j}i_{3-3j}, r_j i_{3-3j}\}$, and *cross circuits* $Q_j = \{\ell_j\ell_{j+1}, \ell_{j+1}r_{j+1}, r_j r_{j+1}, \ell_j r_j\}$ for $j = 0, \ldots, 6$, as well as the triangle $\Delta = \{\ell_0 r_0, r_0 i_0, \ell_0 i_0\}$. Now choose all left spokes, all right spokes, and all inner arcs except $i_6 i_0$ to achieve a spanning tree on D. Then the corresponding reduced cycle matrix Γ' is

$$\begin{array}{cc}
 & \begin{array}{cccc} \text{left circuits} & \text{right circuits} & \text{cross circuits} & \Delta \end{array} \\
\begin{array}{c}
\ell_0\ell_1\\ \ell_1\ell_2\\ \ell_2\ell_3\\ \ell_3\ell_4\\ \ell_4\ell_5\\ \ell_5\ell_6\\ \ell_6\ell_0\\
r_0r_1\\ r_1r_2\\ r_2r_3\\ r_3r_4\\ r_4r_5\\ r_5r_6\\ r_6r_0\\
\ell_1r_1\\ \ell_2r_2\\ \ell_3r_3\\ \ell_4r_4\\ \ell_5r_5\\ \ell_6r_6\\ \ell_0r_0\\
i_6i_0
\end{array} &
\left(\begin{array}{ccccccc|ccccccc|ccccccc|c}
1&&&&&& & &&&&&& & 1&&&&&& & \\
&1&&&&& & &&&&&& & &1&&&&& & \\
&&1&&&& & &&&&&& & &&1&&&& & \\
&&&1&&& & &&&&&& & &&&1&&& & \\
&&&&1&& & &&&&&& & &&&&1&& & \\
&&&&&1& & &&&&&& & &&&&&1& & \\
&&&&&&1 & &&&&&& & &&&&&&1 & \\
\hline
&&&&&& & 1&&&&&& & -1&&&&&& & \\
&&&&&& & &1&&&&& & &-1&&&&& & \\
&&&&&& & &&1&&&& & &&-1&&&& & \\
&&&&&& & &&&1&&& & &&&-1&&& & \\
&&&&&& & &&&&1&& & &&&&-1&& & \\
&&&&&& & &&&&&1& & &&&&&-1& & \\
&&&&&& & &&&&&&1 & &&&&&&-1 & \\
\hline
&&&&&& & &&&&&& & 1&-1&&&&& & \\
&&&&&& & &&&&&& & &1&-1&&&& & \\
&&&&&& & &&&&&& & &&1&-1&&& & \\
&&&&&& & &&&&&& & &&&1&-1&& & \\
&&&&&& & &&&&&& & &&&&1&-1& & \\
&&&&&& & &&&&&& & &&&&&1&-1 & \\
&&&&&& & &&&&&& & -1&&&&&&1 & 1 \\
\hline
1&&&1&&& & 1&&1&&1&& & &&&&&& &
\end{array}\right)
\end{array}.$$

Since $\det(\Gamma') = -1$, which one can see after a few elimination steps, B is an integral cycle basis. Consider now the circuit $C = \{\ell_0\ell_1, \ell_1\ell_2, \ell_2\ell_3, \ell_3\ell_4, \ell_4\ell_5, \ell_5\ell_6, \ell_6\ell_0\}$, i.e. the circuit which consists exactly of all left outer arcs. It has the representation $C = \sum_{j=0}^{6}(3L_j - 2R_j - 2Q_j)$, a representation without a coefficient $+1$ or -1. $\diamond$

5.3.6 Conclusions

This last section treated the practically relevant topic of integral cycle bases. We shortly want to summarize our results and to give some outlooks concerning integral cycle bases.

We gave an example of an integral cycle basis which does not contain simple cycles. This result should help to better understand the structure of integral cycle bases. It may perhaps be of help for answering the question whether there is a (unique) minimum integral cycle basis with a non-circuit—a part of Open Problem 4 in the Survey.

The result in the last subsection could help to further investigate the interaction of integral and totally unimodular cycle bases. Our results and the proposed further research might be helpful for settling the complexity status of the Minimum Integral Cycle Basis Problem.

Bibliography

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS '08*, pages 781–790. IEEE Computer Society, 2008. 53, 64

[2] Yogesh K. Agarwal and Prabha Sharma. Benders' partitioning approach for solving the optimal communication spanning tree problem. In S.K. Neogy, Arup Kumar Das, and Ravi B. Bapat, editors, *Modeling, Computation and Optimization*, Statistical Science and Interdisciplinary Research 6, pages 237–256. World Scientific, 2009. 64

[3] Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK. 4th revised and enlarged ed.* Springer, 2010. 117

[4] Paola Alimonti and Viggo Kann. Hardness of approximating problems on cubic graphs. In Gian Carlo Bongiovanni, Daniel P. Bovet, and Giuseppe Di Battista, editors, *Algorithms and Complexity, Third Italian Conference, CIAC '97*, LNCS 1203, pages 288–298. Springer, 1997. 27, 42

[5] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, 1990. 81

[6] Edoardo Amaldi, Leo Liberti, Francesco Maffioli, and Nelson Maculan. Edge-swapping algorithms for the minimum fundamental cycle basis problem. *Mathematical Methods of Operations Research*, 69(1):205–233, 2009. 57, 85

[7] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and Approximation. Combinatorial Optimization Problems and Their Approximability Properties.* Springer, 1999. 23, 27

[8] Giorgio Ausiello and Vangelis Th. Paschos. Approximation preserving reductions. In Vangelis Th. Paschos, editor, *Paradigms of Combinatorial Optimization. Problems and New Approaches*, Combinatorial Optimization. Volume 2, pages 351–380. John Wiley & Sons, 2010. 27

[9] Simone Bächle and Falk Ebert. Graph theoretical algorithms for index reduction in circuit simulation. Technical Report 245, DFG Research Center MATHEON, 2005. 54

[10] Alexandru T. Balaban. A trivalent graph of girth ten. *Journal of Combinatorial Theory (B)*, 12:1–5, 1972. 119

[11] Balabhaskar Balasundaram and Sergiy Butenko. Graph domination, coloring and cliques in telecommunications. In Mauricio G.C. Resende and Panos M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, pages 865–890. Springer, 2006. 32

[12] Claude Berge. *The Theory of Graphs and its Applications.* John Wiley & Sons, 1962. 123

[13] Franziska Berger, Christoph Flamm, Petra M. Gleiss, Josef Leydold, and Peter F. Stadler. Counterexamples in chemical ring perception. *Journal of Chemical Information and Computer Sciences*, 44:323–331, 2004. 52, 55, 95

[14] Randeep Bhatia, Samir Khuller, Robert Pless, and Yoram J. Sussmann. The full-degree spanning tree problem. *Networks*, 36(4):203–209, 2000. 50

[15] Daniel Binkele-Raible and Henning Fernau. An exact exponential-time algorithm for the directed maximum leaf spanning tree problem. *Journal of Discrete Algorithms*, 15:43–55, 2012. 31

[16] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998. 49

[17] Béla Bollobás. *Modern Graph Theory.* Springer, 1998. 15, 54, 116

[18] John A. Bondy and U.S.R. Murty. *Graph Theory.* Springer, 2008. 15, 21, 54, 137

[19] Paul Bonsma. Max-leaves spanning tree is APX-hard for cubic graphs. *Journal of Discrete Algorithms*, 12:14–23, 2012. 29, 30, 31

[20] Paul Bonsma and Florian Zickfeld. A 3/2-approximation algorithm for finding spanning trees with many leaves in cubic graphs. In Hajo Broersma, Thomas Erlebach, Tom Friedetzky, and Daniel Paulusma, editors, *Graph-Theoretic Concepts in Computer Science*, LNCS 5344, pages 66–77. Springer, 2008. 29, 30, 50

[21] Hajo Broersma, Otto Koppius, Hilde Tuinstra, Andreas Huck, Ton Kloks, Dieter Kratsch, and Haiko Müller. Degree-preserving trees. *Networks*, 35(1):26–39, 2000. 48, 49

[22] Arthur Cayley. On the analytic forms called trees, with applications to the theory of chemical combinations. *Reports of the British Association for the Advancement of Science*, 45:257–305, 1875. 54

[23] Christophe Champetier. On the null-homotopy of graphs. *Discrete Mathematics*, 64:97–98, 1987. 110

[24] Gary Chartrand, Dennis Geller, and Stephen Hedetniemi. Graphs with forbidden subgraphs. *Journal of Combinatorial Theory (B)*, 10:2–41, 1971. 82

[25] Gary Chartrand and Frank Harary. Planar permutation graphs. *Annales de l'Institut Henri Poincaré. Nouvelle Série. Section B*, 3:433–438, 1967. 81

[26] Si Chen, Ivana Ljubić, and Srinivasacharya Raghavan. The regenerator location problem. *Networks*, 55(3):205–220, 2010. 32, 33

[27] *Complexity Zoo*. `http://qwiki.stanford.edu/index.php/Complexity_Zoo` (Last accessed April 24, 2014). 24

[28] José R. Correa, Cristina G. Fernandes, Martín Matamala, and Yoshiko Wakabayashi. A 5/3-approximation for finding spanning trees with many leaves in cubic graphs. In Christos Kaklamanis and Martin Skutella, editors, *Approximation and Online Algorithms*, LNCS 4927, pages 184–192. Springer, 2008. 29, 30, 31, 50

[29] Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity, CCC '97*, pages 262–273. IEEE Computer Society, 1997. 27

[30] Pierluigi Crescenzi, Viggo Kann, Riccardo Silvestri, and Luca Trevisan. Structure in approximation classes. *SIAM Journal on Computing*, 28(5):1759–1782, 1999. 25

[31] Narsingh Deo and Nishit Kumar. Computation of constrained spanning trees: A unified approach. In Panos M. Pardalos, Donald W. Hearn, and William W. Hager, editors, *Network Optimization*, Lecture Notes in Economics and Mathematical Systems 450, pages 194–220. Springer, 1997. 64

[32] Narsingh Deo, Gurpur M. Prabhu, and Mukkai S. Krishnamoorthy. Algorithms for generating fundamental cycles in a graph. *ACM Transactions on Mathematical Software*, 8:26–42, 1982. 51, 52, 121

[33] Giuseppe Di Battista and Roberto Tamassia. Incremental planarity testing. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science, FOCS '89*, pages 436–441. IEEE Computer Society, 1989. 61

[34] Pablo Diaz-Gutierrez, Anusheel Bhushan, Meenakshisundaram Gopi, and Renato Pajarola. Single-strips for fast interactive rendering. *The Visual Computer*, 22(6):372–386, 2006. 31

[35] *Dictionary.com*. `http://dictionary.reference.com` (Last accessed April 24, 2014). 11

[36] Reinhard Diestel. *Graph Theory.* Springer, 2008. 15, 60, 61, 116, 117

[37] E.T. Dixon and Seymour E. Goodman. An algorithm for the longest cycle problem. *Networks*, 6:139–149, 1976. 94

[38] Ugur Dogrusöz and Mukkai S. Krishnamoorthy. Cycle vector space algorithms for enumerating all cycles of a planar graph. *The Journal of Parallel Algorithms and Applications*, 10:21–36, 1996. 97, 106

[39] Geoffrey M. Downs, Valerie J. Gillet, John D. Holliday, and Michael F. Lynch. Review of ring perception algorithms for chemical graphs. *Journal of Chemical Information and Computer Sciences*, 29:187–206, 1989. 95

[40] Pierre Duchet, Michel Las Vergnas, and Henry Meyniel. Connected cutsets of a graph and triangle bases of the cycle space. *Discrete Mathematics*, 62:145–154, 1986. 110

[41] Martin E. Dyer and Alan M. Frieze. Planar 3DM is NP-complete. *Journal of Algorithms*, 7:174–184, 1986. 34, 35, 69

[42] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2009. 63, 64

[43] Geoffrey Exoo and Robert Jajcay. Dynamic cage survey. *The Electronic Journal of Combinatorics*, 18, 2011. 118, 119

[44] Henning Fernau, Joachim Kneis, Dieter Kratsch, Alexander Langer, Mathieu Liedloff, Daniel Raible, and Peter Rossmanith. An exact algorithm for the maximum leaf spanning tree problem. *Theoretical Computer Science*, 412(45):6290–6302, 2011. 30

[45] Rudolf Fleischer and Colin Hirsch. Graph drawing and its applications. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs. Methods and Models*, LNCS 2025, pages 1–22. Springer, 2001. 53

[46] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. Solving connected dominating set faster than 2^n. *Algorithmica*, 52(2):153–166, 2008. 30

[47] Greg N. Frederickson. Planar graph decomposition and all pairs shortest paths. *Journal of the Association for Computing Machinery*, 38(1):162–204, 1991. 81

[48] Greg N. Frederickson. Using cellular graph embeddings in solving all pairs shortest paths problems. *Journal of Algorithms*, 19(1):45–85, 1995. 81

[49] Thomas M.J. Fruchtermann and Edward M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991. 54

[50] Tetsuya Fujie. An exact algorithm for the maximum leaf spanning tree problem. *Computers & Operation Research*, 30(13):1931–1944, 2003. 30

[51] Giulia Galbiati, Francesco Maffioli, and Angelo Morzenti. A short note on the approximability of the maximum leaves spanning tree problem. *Information Processing Letters*, 52(1):45–49, 1994. 30, 41, 42

[52] Giulia Galbiati, Romeo Rizzi, and Edoardo Amaldi. On the approximability of the minimum strictly fundamental cycle basis problem. *Discrete Applied Mathematics*, 159(4):187–200, 2011. 34, 53, 121

[53] Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979. 22, 29, 30, 33, 34, 64

[54] Petra M. Gleiss. *Short Cycles.* PhD thesis, Universität Wien, 2001. 55, 61

[55] Chris Godsil and Gordon Royle. *Algebraic Graph Theory.* Springer, 2001. 21

[56] Daniel Gonçalves. Edge partition of planar graphs into two outerplanar graphs. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 504–512. Association for Computing Machinery, 2005. 82

[57] David Hartvigsen and Eitan Zemel. Is every cycle basis fundamental? *Journal of Graph Theory*, 13(1):117–137, 1989. 57

[58] Lenwood S. Heath. Embedding outerplanar graphs in small books. *SIAM Journal on Algebraic and Discrete Methods*, 8:198–218, 1987. 81

[59] John E. Hopcroft, Jeffrey D. Ullman, and Rajeev Motwani. *Introduction to Automata Theory, Languages, and Computation. 3rd ed.* Addison-Wesley, 2007. 22

[60] Tamás Horváth, Jan Ramon, and Stefan Wrobel. Frequent subgraph mining in outerplanar graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 197–206. Springer, 2006. 81

[61] T.C. Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3:188–195, 1974. 64

[62] David S. Johnson, Jan K. Lenstra, and Alexander H.G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8:279–285, 1978. 34

[63] Paul C. Kainen. On robust cycle bases. *Electronic Notes in Discrete Mathematics*, 11:430–437, 2002. 94, 95, 97, 100, 104

[64] Telikepalli Kavitha, Christian Liebchen, Kurt Mehlhorn, Dimitrios Michail, Romeo Rizzi, Torsten Ueckerdt, and Katharina A. Zweig. Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review*, 3(4):199–243, 2009. 13, 20, 51, 52, 57, 61, 87, 115, 121, 122, 123, 124, 127

[65] Gustav R. Kirchhoff. Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der Linearen Vertheilung galvanischer Ströme geführt wird. *Annalen der Physik und Chemie*, 72:497–508, 1847. 52, 56

[66] Konstantin Klemm and Peter F. Stadler. Statistics of cycles in large networks. *Physical Review E*, 73:025101, 2006. 94

[67] Konstantin Klemm and Peter F. Stadler. A note on fundamental, non-fundamental, and robust cycle bases. *Discrete Applied Mathematics*, 157(10):2432–2438, 2009. 13, 52, 93, 94, 96, 97, 104, 105, 106, 110, 111, 112

[68] Ekkehard Köhler, Christian Liebchen, Gregor Wünsch, and Romeo Rizzi. Lower bounds for strictly fundamental cycle bases in grid graphs. *Networks*, 53(2):191–205, 2009. 53, 54

[69] W. Koontz. Economic evaluation of loop feeder relief alternatives. *The Bell System Technical Journal*, 59:277–281, 1980. 81

[70] Sven Oliver Krumke and Hartmut Noltemeier. *Graphentheoretische Konzepte und Algorithmen.* Teubner, 2005. 59

[71] Jan van Leeuwen, editor. *Algorithms and Complexity. Handbook of Theoretical Computer Science. Vol. A*, chapter 10. Elsevier Science Publishers, 1990. 22, 81

[72] Katharina A. Lehmann and Stephan Kottler. Visualizing large and clustered networks. In Michael Kaufmann and Dorothea Wagner, editors, *Graph Drawing*, LNCS 4372, pages 240–251. Springer, 2007. 54

[73] Paul Lemke. The maximum leaf spanning tree problem for cubic graphs is NP-complete. Technical Report IMA publication no. 428, University of Minnesota, 1988. 29, 30, 34, 35, 37, 39

[74] Josef Leydold and Peter F. Stadler. Minimal cycle bases of outerplanar graphs. *The Electronic Journal of Combinatorics*, 5:209–222, 1998. 81, 82

[75] P.C. Li and Michel Toulouse. Maximum leaf spanning tree problem for grid graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 73:181–193, 2010. 31

[76] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982. 34

[77] Christian Liebchen. Finding short integral cycle bases for cyclic timetabling. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003*, LNCS 2832, pages 715–726. Springer, 2003. 133

[78] Christian Liebchen. *Periodic Timetable Optimization in Public Transport.* PhD thesis, TU Berlin, 2006. 51, 52, 56, 57, 61, 66, 68, 105, 106, 108, 109, 110, 111, 115, 120, 122, 123, 124, 125, 127, 130, 133, 134

[79] Christian Liebchen and Leon Peeters. On cyclic timetabling and cycles in graphs. Technical Report 761-2002, TU Berlin, 2002. 123, 133, 134, 136

[80] Christian Liebchen and Leon Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 6(1):98–109, 2008. 133

[81] Christian Liebchen and Romeo Rizzi. Classes of cycle bases. *Discrete Applied Mathematics*, 155(3):337–355, 2007. 61, 114

[82] Christian Liebchen, Gregor Wünsch, Ekkehard Köhler, Alexander Reich, and Romeo Rizzi. Benchmarks for strictly fundamental cycle bases. In Camil Demetrescu, editor, *Experimental Algorithms - WEA 2007*, LNCS 4525, pages 365–378. Springer, 2007. 53

[83] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36:177–189, 1979. 81

[84] Krzysztof Loryś and Grażyna Zwoźniak. Approximation algorithm for the maximum leaf spanning tree problem for cubic graphs. In Rolf Möhring and Rajeev Raman, editors, *Algorithms - ESA 2002*, LNCS 2461, pages 686–697. Springer, 2002. 30, 31

[85] Hsueh-i Lu and Ramamurthy Ravi. Approximating maximum leaf spanning trees in almost linear time. *Journal of Algorithms*, 29(1):132–141, 1998. 30

[86] Jin Lu and Josée M.F. Moura. Structured LDPC codes for high-density recording: Large girth and low error floor. *IEEE transactions on magnetics*, 42:208–213, 2006. 118

[87] Saunders MacLane. A combinatorial condition for planar graphs. *Fundamenta Mathematicae*, 28:22–32, 1937. 61, 117

[88] Jean F. Maurras, Klaus Truemper, and Mustafa Akguel. Polynomial algorithms for a class of linear programs. *Mathematical Programming*, 21:121–136, 1981. 121

[89] Kurt Mehlhorn and Stefan Näher. *LEDA. A platform for combinatorial and geometric computing.* Cambridge University Press, 1999. 110

[90] Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927. 116

[91] Keizo Miyata, Shigeru Masuyama, Shin-Ichi Nakayama, and Liang Zhao. NP-hardness proof and an approximation algorithm for the minimum vertex ranking spanning tree problem. *Discrete Applied Mathematics*, 154(16):2402–2410, 2006. 34

[92] Karl Nachtigall. A branch and cut approach for periodic network programming. Technical Report 29, Hildesheimer Informatik-Berichte, 1994. 135

[93] *NCI dataset.* http://cactus.nci.nih.gov/ (Last accessed April 24, 2014). 81

[94] Tetsuo Nishi and Leon O. Chua. Uniqueness of solution for nonlinear resistive circuits containing CCCS's or VCVS's whose controlling coefficients are finite. *IEEE Transactions on Circuits and Systems*, 33(4):381–397, 1986. 81

[95] M. O'Keefe and Pak-Ken Wong. A smallest graph of girth 10 and valency 3. *Journal of Combinatorial Theory (B)*, 29:91–105, 1980. 119

[96] *Online Etymology Dictionary.* http://www.etymonline.com (Last accessed April 24, 2014). 11

[97] Øystein Ore. *Theory of Graphs.* American Mathematical Society (AMS), 1962. 30

[98] Philipp-Jens Ostermeier. *Quasi-robust Cycle Spaces.*, 2009. http://www.bioinf.uni-leipzig.de/conference-registration/09herbst/talks/401_Ostermeier.pdf. 94, 101

[99] Philipp-Jens Ostermeier, Marc Hellmuth, Josef Leydold, Konstantin Klemm, and Peter F. Stadler. A note on quasi-robust cycle bases. *Ars Mathematica Contemporanea*, 2(2):231–240, 2009. 93, 94, 96, 97, 100

[100] Leon Peeters. *Cyclic Railway Timetable Optimization.* PhD thesis, Erasmus University Rotterdam, 2003. 134

[101] David Peleg and Dov Tendler. Low stretch spanning trees for planar graphs. Technical Report MCS01-14, Weizmann Institute of Science, 2001. 62, 83, 86, 90

[102] José de Pina. *Applications of Shortest Path Methods.* PhD thesis, Universiteit van Amsterdam, 2005. 68

[103] Ivo W.M. Pothof and Jan Schut. Graph-theoretic approach to identifiability in a water distribution network. Memorandum 1283, Faculty of Applied Mathematics, University of Twente, Enschede, 1995. 48

[104] Boto von Querenburg. *Mengentheoretische Topologie. 3. Auflage.* Springer, 2001. 59

[105] Jan Ramon, Tamás Horváth, Leander Schietgat, and Stefan Wrobel. FOG: Finding outerplanar graphs. In G. Melli, editor, *Online Proceedings of the Demo Session of the ACM SIGKDD Conference 2006*, pages 1–3, 2006. 81

[106] Ronald C. Read. On general dissections of a polygon. *Aequationes Mathematicae*, 18:370–388, 1978. 81

[107] Alexander Reich. The maximum leaf spanning tree problem on planar and regular graphs. Submitted. 29

[108] Alexander Reich. Streng fundamentale Kreisbasen planarer Graphen. Master's thesis, BTU Cottbus, 2007. 52, 81, 84, 133, 134

[109] Franz Rothlauf. On optimal solutions for the optimal communication spanning tree problem. *Operations Research*, 57(2):413–425, 2009. 64

[110] Horst Sachs. Regular graphs with given girth and restricted circuits. *Journal of the London Mathematical Society*, 38:423–429, 1963. 118

[111] E. Sampathkumar and Hanumappa B. Walikar. The connected domination number of a graph. *Journal of Mathematical and Physical Sciences*, 13:607–613, 1979. 30

[112] Norbert Sauer. Extremaleigenschaften regulärer Graphen gegebener Taillenweite, I und II. *Österreichische Akademie der Wissenschaften, Mathematisch-Naturwissenschaftliche Klasse. Sitzungsberichte. Abteilung II*, 176:9–43, 1967. 119

[113] Alexander Schrijver. *Theory of Linear and Integer Programming. Reprint.* Chichester: Wiley, 1998. 121, 122

[114] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989. 133

[115] Prabha Sharma. Algorithms for the optimum communication spanning tree problem. *Annals of Operations Research*, 143:203–209, 2006. 64

[116] Paul L. Shick. *Topology. Point-Set and Geometric.* Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs, and Tracts, 2007. 59

[117] Ana Silva, Aline Alves da Silva, and Cláudia Linhares Sales. A bound on the treewidth of planar even-hole-free graphs. *Discrete Applied Mathematics*, 158(12):1229–1239, 2010. 81

[118] Roberto Solis-Oba. 2-approximation algorithm for finding a spanning tree with maximum number of leaves. In Gianfranco Bilardi, Giuseppe F. Italiano, Andrea Pietracaprina, and Geppino Pucci, editors, *Algorithms - ESA '98*, LNCS 1461, pages 24–26. Springer, 1998. 30, 45

[119] Maciej Marek Sysło. On cycle bases of a graph. *Networks*, 9:123–132, 1979. 94, 101

[120] Maciej Marek Sysło. On some problems related to fundamental cycle sets of a graph: research notes. *Discrete Mathematics*, 7:145–157, 1982. 57

[121] James R. Walter. Representations of chordal graphs as subtrees of a tree. *Journal of Graph Theory*, 2:265–267, 1978. 49

[122] Michael S. Waterman. Secondary structure of single-stranded nucleic acids. In *Studies on Foundations and Combinatorics*, Advances in Mathematics Supplementary Studies, Volume 1, pages 167–212. Academic Press, 1978. 81

[123] Douglas B. West. *Introduction to Graph Theory. 2nd ed.* Prentice-Hall, 2005. 15

[124] Hassler Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34:339–362, 1932. 116

[125] Hassler Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–254, 1933. 60

[126] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57:509–533, 1935. 52, 55

[127] Pak-Ken Wong. On the smallest graphs of girth 10 and valency 3. *Discrete Mathematics*, 43:119–124, 1983. 119

[128] Bang Ye Wu, Giuseppe Lancia, Vineet Bafna, Kun-Mao Chao, and Ramamurthy Ravi. A polynomial-time approximation scheme for minimum routing cost spanning trees. *SIAM Journal on Computing*, 29(3):761–778, 2000. 64, 65

[129] Gregor Wünsch. *Coordination of Traffic Signals in Networks and Related Graph Theoretical Problems on Spanning Trees.* PhD thesis, TU Berlin, 2008. 51, 66

[130] Hitoshi Yamasaki, Yosuke Sasaki, Takayoshi Shoudai, Tomoyuki Uchida, and Yusuke Suzuki. Learning block-preserving outerplanar graph patterns and its application to data mining. In Filip Železný and Nada Lavrač, editors, *Inductive Logic Programming*, LNCS 5194, pages 330–347. Springer, 2008. 81

[131] *Zentralblatt MATH.* `http://www.zentralblatt-math.org` (Last accessed April 24, 2014). 94

Index

List of Problems

In this thesis appear many decision and optimization problems, and often they are abbreviated. This list summarizes the used problems in the alphabetic order of their abbreviations. The problems in the first group on this page are defined in a stringent manner, while the second group on the next page lists up further problems which occur in this thesis.